



Universidad
Carlos III de Madrid

DEPARTAMENTO DE INFORMÁTICA

PROYECTO FIN DE CARRERA

Ingeniería en Informática

“UN ALGORITMO GENÉTICO CON CODIFICACIÓN REAL PARA LA EVOLUCIÓN DE TRANSFORMACIONES LINEALES”

Autor: José Carlos López Díaz

Tutores: Ricardo Aler Mur / José María Valls Ferrán

Leganés, Julio de 2010

A mis padres y hermanos

A Ara

Índice de contenido

1.	Introducción.....	14
1.1.	Resumen y objetivos del proyecto	14
1.2.	Estructura del proyecto	16
1.2.1.	Capítulo 2: Algoritmos Genéticos	16
1.2.2.	Capítulo 3: Algoritmos de clasificación. Distancias y proyecciones.	16
1.2.3.	Capítulo 4: Sistema implementado.....	16
1.2.4.	Capítulo 5: Experimentación	16
1.2.5.	Capítulo 6: Conclusiones	17
2.	Algoritmos Genéticos	19
2.1.	¿Que son los algoritmos genéticos?	19
2.2.	Antecedentes históricos	20
2.3.	Características	21
2.4.	Representación.....	23
2.5.	Algoritmo	25
2.6.	Función de evaluación y función de aptitud (fitness)	26
2.7.	Operadores genéticos.....	27
2.7.1.	Selección.....	27
2.7.2.	Reproducción.....	27
2.7.3.	Cruce	28
2.7.4.	Mutación	29
3.	Algoritmos de Clasificación. Distancias y proyecciones	32
3.1.	Introducción	32
3.2.	Aprendizaje inductivo	34
3.3.	Algoritmos de clasificación por vecindad.....	35
3.3.1.	Algoritmo KNN.....	35
3.4.	Distancias	37
3.4.1.	Distancia euclídea no ponderada	37
3.4.2.	Distancia euclídea ponderada en diagonal	38
3.4.3.	Distancia euclídea plenamente ponderada.....	38
3.5.	Proyecciones	40
4.	Sistema Implementado.....	42
4.1.	Objetivos y descripción del sistema	42

4.1.1.	Objetivo del proyecto.....	42
4.1.2.	Definición del algoritmo genético.....	42
4.1.3.	Función de Fitness	45
4.2.	Análisis del sistema.....	46
4.2.1.	Requisitos no funcionales.....	46
4.2.2.	Requisitos funcionales.....	46
4.3.	Funcionamiento global del sistema	49
4.4.	Diseño del sistema.....	51
4.4.1.	Paquete “main”	52
4.4.2.	Paquete “model”	52
4.4.3.	Paquete “control.util”	53
4.4.4.	Paquete “control.knn”	54
4.4.5.	Paquete “control.ga”.....	55
5.	Experimentación	57
5.1.	Dominios utilizados en la experimentación	57
5.2.	Ripley	58
5.2.1.	Descripción del dominio.....	58
5.2.2.	Configuración.....	59
5.2.3.	Resultados	59
5.3.	Diabetes	61
5.3.1.	Descripción del dominio.....	61
5.3.2.	Configuración.....	62
5.3.3.	Resultados	62
5.4.	Rectas Ruido 0 grados.....	64
5.4.1.	Descripción del dominio.....	64
5.4.2.	Configuración.....	65
5.4.3.	Resultados	65
5.5.	Rectas Ruido 45 grados.....	67
5.5.1.	Descripción del dominio.....	67
5.5.2.	Configuración.....	68
5.5.3.	Resultados	69
5.6.	Rectas4D-45g	71
5.6.1.	Descripción del dominio.....	71
5.6.2.	Configuración.....	72

5.6.3.	Resultados	72
5.7.	Aleatorio	74
5.7.1.	Descripción del dominio.....	74
5.7.2.	Configuración.....	75
5.7.3.	Resultados	75
5.8.	Aleatorio Girado	77
5.8.1.	Descripción del dominio.....	77
5.8.2.	Configuración.....	77
5.8.3.	Resultados	78
5.9.	EEG.....	80
5.9.1.	Descripción del dominio.....	80
5.9.2.	Configuración.....	82
5.9.3.	Resultados	83
6.	Conclusiones	87
ANEXO A:	Matrices Resultado	90
1.	Matrices resultado del dominio Ripley.....	90
1.1.	Matriz Simétrica	90
1.2.	Matriz Diagonal	91
1.3.	Matriz Completa	91
2.	Matrices resultado del dominio Diabetes.....	93
2.1.	Matriz Simétrica	93
2.2.	Matriz Diagonal	94
2.3.	Matriz Completa	94
3.	Matrices resultado del dominio Rectas Ruido 0 grados	96
3.1.	Matriz Simétrica	96
3.2.	Matriz Diagonal	97
3.3.	Matriz Completa	98
4.	Matrices resultado del dominio Rectas Ruido 45 grados	100
4.1.	Matriz Simétrica	100
4.2.	Matriz Diagonal	101
4.3.	Matriz Completa	102
5.	Matrices resultado del dominio Rectas4D-45g.....	104
5.1.	Matriz Simétrica	104
5.2.	Matriz Diagonal	105

5.3.	Matriz Completa	105
6.	Matrices resultado del dominio Aleatorio.....	107
6.1.	Matriz Simétrica	107
6.2.	Matriz Diagonal	108
6.3.	Matriz Completa	108
7.	Matrices resultado del dominio Aleatorio Girado	110
7.1.	Matriz Simétrica	110
7.2.	Matriz Diagonal	111
7.3.	Matriz Completa	111
8.	Matrices resultado del dominio EEG	113
8.1.	Conjunto de datos con clases 2 y 3.....	113
8.2.	Conjunto de datos con clases 2 y 7.....	114
8.3.	Conjunto de datos con clases 3 y 7.....	115
ANEXO B: Presupuesto.....		117
1.	Costes de Personal	117
2.	Costes de Material.....	117
3.	Presupuesto Final	118
Referencias		119

Índice de Ilustraciones

Ilustración 1: Charles Darwin.....	20
Ilustración 2: Cromosoma	23
Ilustración 3: Ejemplo de cromosoma de representación binaria.....	23
Ilustración 4: Ejemplo de cromosoma de representación entera	24
Ilustración 5: Ejemplo de cromosoma de representación real	24
Ilustración 6: Ejemplo de cruce por un punto	28
Ilustración 7: Ejemplo de cruce multipunto de dos puntos	29
Ilustración 8: Ejemplo de mutación	30
Ilustración 9: Ejemplo algoritmo KNN.....	36
Ilustración 10: Ejemplo de función de distancia euclídea no ponderada	37
Ilustración 11: Ejemplo de función de distancia euclídea ponderada diagonalmente	38
Ilustración 12: Ejemplo de función de distancia euclídea plenamente ponderada	39
Ilustración 13: Proyección de un dato	40
Ilustración 14: Diagrama de clases JGAP.....	43
Ilustración 15: Flujo de ejecución	50
Ilustración 16: Diagrama de clases de paquetes	51
Ilustración 17: Clases del paquete main	52
Ilustración 18: Clases del paquete model	52
Ilustración 19: Clases del paquete control.util	53
Ilustración 20: Clases del paquete control.knn	54
Ilustración 21: Clases del paquete control.ga	55
Ilustración 22: Datos del dominio Ripley.....	58
Ilustración 23: Resultados del dominio Ripley.....	60
Ilustración 24: Resultados del dominio Diabetes	63
Ilustración 25: Datos del dominio Rectas ruido 0 grados.....	64
Ilustración 26: Resultados del dominio Rectas ruido 0 grados.....	66
Ilustración 27: Datos del dominio Rectas ruido 45 grados.....	67
Ilustración 28: Datos del dominio Rectas ruido 45 grados.....	67
Ilustración 29: Resultados del dominio Rectas ruido 45 grados.....	70
Ilustración 30: Resultados del dominio Rectas 4D 45 grados.....	73
Ilustración 31: Resultados del dominio Aleatorio.....	76
Ilustración 32: Resultados del dominio Aleatorio Girado	79
Ilustración 33: Datos del dominio EEG.....	80
Ilustración 34: Resultados del dominio EEG con clases 2 y 3	83
Ilustración 35: Resultados del dominio EEG con clases 2 y 7	84
Ilustración 36: Resultados del dominio EEG con clases 3 y 7	85

Índice de Tablas

Tabla 1: Características del dominio Ripley.....	58
Tabla 2: Resultados del dominio Ripley	59
Tabla 3: Características del dominio Diabetes.....	61
Tabla 4: Resultados del dominio Diabetes	62
Tabla 5: Características del dominio Rectas ruido 0 grados.....	64
Tabla 6: Resultados del dominio Rectas ruido 0 grados	66
Tabla 7: Características del dominio Rectas Ruido 45 grados	68
Tabla 8: Resultados del dominio Rectas ruido 45 grados	69
Tabla 9: Características del dominio Rectas4D-45g.....	71
Tabla 10: Resultados del dominio Rectas 4D 45 grados	72
Tabla 11: Características del dominio Aleatorio.....	74
Tabla 12: Resultados del dominio Aleatorio.....	75
Tabla 13: Características del dominio Aleatorio Girado.....	77
Tabla 14: Resultados del dominio Aleatorio Girado	78
Tabla 15: Características del dominio EEG con clases 2 y 3	80
Tabla 16: Características del dominio EEG con clases 2 y 7	81
Tabla 17: Características del dominio EEG con clases 3 y 7	82
Tabla 18: Resultados del dominio EEG con clases 2 y 3.....	83
Tabla 19: Resultados del dominio EEG con clases 2 y 7.....	84
Tabla 20: Resultados del dominio EEG con clases 3 y 7.....	84
Tabla 21: Resumen de resultados.....	87
Tabla 22: Resultados del dominio Ripley	90
Tabla 23: Resultados del dominio Diabetes	93
Tabla 24: Resultados del dominio Rectas Ruido 0 grados.....	96
Tabla 25: Resultados del dominio Rectas Ruido 45 grados.....	100
Tabla 26: Resultados del dominio Rectas4D 45 grados	104
Tabla 27: Resultados del dominio Aleatorio.....	107
Tabla 28: Resultados del dominio Aleatorio Girado	110
Tabla 29: Resultados del dominio EEG con clases 2 y 3.....	113
Tabla 30: Resultados del dominio EEG con clases 2 y 7.....	114
Tabla 31: Resultados del dominio EEG con clases 3 y 7.....	115
Tabla 32: Personal.....	117
Tabla 33: Costes de Personal.....	117
Tabla 34: Costes de Material.....	118
Tabla 35: Presupuesto Final	118

Índice de Ecuaciones

Ecuación 1: Distancia euclídea ponderada.....	14
Ecuación 2: Transformación lineal de los datos	14
Ecuación 3: Fórmula de la distancia Euclídea no ponderada	37
Ecuación 4: Fórmula de la distancia Euclídea ponderada en diagonal	38
Ecuación 5: Fórmula distancia euclídea plenamente ponderada.....	39
Ecuación 6: Distancia euclídea ponderada.....	42
Ecuación 7: Fórmula de la distancia euclídea ponderada	54

CAPÍTULO 1: INTRODUCCIÓN

En este primer capítulo se exponen los objetivos generales hacia los que va dirigido este proyecto, así como un breve resumen del mismo. También se detallan brevemente los contenidos de cada uno de los capítulos del proyecto.

1. Introducción

1.1. Resumen y objetivos del proyecto

Los *algoritmos de clasificación* por vecindad pertenecen al grupo de las técnicas de clasificación supervisadas, que nos permiten obtener un modelo para clasificar casos futuros. Los métodos de clasificación exigen la definición de una cierta medida de disimilitud (o distancia) entre los distintos elementos, es decir, precisan de una métrica que ayude a comparar las distancias entre los distintos objetos.

Los *algoritmos genéticos* (AGs) son mecanismos de búsqueda basados en las leyes de la selección natural y de la genética. Combinan la supervivencia de los individuos mejor adaptados junto con operadores de búsqueda genéticos como la mutación y el cruce, de ahí que sean comparables a una búsqueda biológica. Estos algoritmos se utilizan con éxito para gran variedad de problemas de optimización y que no permiten una solución eficiente a través de la aplicación de técnicas convencionales.

El principal objetivo de este proyecto es diseñar y desarrollar un sistema que utilice un algoritmo genético para optimizar funciones de distancia para el algoritmo de clasificación K-NN. Normalmente, K-NN utilizar la distancia euclídea independientemente del dominio, pero en este proyecto se pretende encontrar la distancia más apropiada para cada dominio. Este proceso es igual a encontrar la matriz M que define la función de distancia d en la ecuación 1 (a estas distancias se las denomina distancias euclídeas generalizadas). En caso de que la matriz M sea la identidad, la distancia d se convierte en la distancia euclídea:

$$d(A, B) \equiv \sqrt{(A - B)^T M^T M (A - B)}$$

Ecuación 1: Distancia euclídea ponderada

Una distancia euclídea generalizada es equivalente a una transformación lineal de los datos, como demuestra la siguiente ecuación, donde se ve que $d(A, B)$ es equivalente a una distancia euclídea entre $A' = MA$ y $B' = MB$, una vez que se han transformado los datos mediante la matriz M :

$$d(A, B) \equiv \sqrt{(MA - MB)^T (MA - MB)} = \sqrt{(A' - B')^T (A' - B')}$$

Ecuación 2: Transformación lineal de los datos

Una vez implementado el sistema, se realizarán experimentaciones con distintos dominios de datos, para comprobar si efectivamente se encuentran matrices que

permitan a K-NN encontrar porcentajes de aciertos mejores que si se hubiera usado la distancia euclídea normal.

Se compararan los resultados que obtiene K-NN utilizando la distancia euclídea, con los resultados que se obtienen utilizando la función anterior, ponderada con matrices evolucionadas por el algoritmo genético que serán de distintos tipos: matrices completas, simétricas y diagonales.

1.2. Estructura del proyecto

A continuación se detallan de forma breve los contenidos de cada uno de los capítulos del proyecto:

1.2.1. Capítulo 2: Algoritmos Genéticos

Este capítulo constituye una introducción a los *Algoritmos Genéticos*, utilizados como sistema de búsqueda y optimización, que forma una parte fundamental del proyecto.

Se detallaran las generalidades de los algoritmos genéticos, así como su historia y la equivalencia de la terminología que se usa respecto a la usada de forma habitual en la genética biológica. A continuación se detallan los distintos tipos de codificación que se pueden utilizar, así como una descripción general de los operadores genéticos

1.2.2. Capítulo 3: Algoritmos de clasificación. Distancias y proyecciones.

A lo largo de este capítulo hablaremos de los algoritmos de clasificación, los diferentes tipos que existen y entraremos un poco más en detalle en el algoritmo de clasificación utilizado para la realización de este proyecto.

Además, hablaremos de las distancias y proyecciones, las cuales son utilizadas por gran parte de los algoritmos de este tipo.

1.2.3. Capítulo 4: Sistema implementado

En este capítulo se describe todo el trabajo realizado en el proyecto.

Se comienza el análisis del sistema en el que se describe el objetivo del proyecto y la configuración del algoritmo de clasificación y del algoritmo genético utilizados. A continuación se describen los requisitos y el funcionamiento global del sistema. Se sigue con el diseño del sistema, en el que se decide cómo llevar a cabo el trabajo descrito en dicho análisis

1.2.4. Capítulo 5: Experimentación

El valor aportado por el proyecto se ha obtenido en la fase de experimentación, en la que se han ido obteniendo resultados con el sistema implementado.

En esta fase del desarrollo del proyecto se han utilizado diferentes dominios de datos para probar el funcionamiento del programa y poder obtener finalmente unos resultados. Los resultados obtenidos nos ayudarán a extraer conclusiones acerca de la combinación

del algoritmo de clasificación KNN y de algoritmos genéticos para encontrar la matriz de distancias más adecuada que nos permita obtener las mejores tasas de aciertos en la clasificación de datos desconocidos.

1.2.5. Capítulo 6: Conclusiones

En este capítulo se detallaran las conclusiones a las que se han llegado tras la realización de este proyecto.

Además de estos contenidos, se añaden dos anexos con la siguiente información:

- ANEXO A: Que contiene las matrices resultado de las experimentaciones llevadas a cabo en este proyecto
- ANEXO B: Que contiene el presupuesto de coste que se realizó para este proyecto.

CAPÍTULO 2: ALGORITMOS GENÉTICOS

Este capítulo constituye una introducción a los algoritmos genéticos, utilizados como sistema de búsqueda y optimización, que forma una parte fundamental del proyecto.

Se detallaran las generalidades de los algoritmos genéticos, así como su historia y la equivalencia de la terminología que se usa respecto a la usada de forma habitual en la genética biológica.

Se pretende aclarar los conceptos generales que se utilizarán más adelante en el proyecto. Para una introducción más detallada, el lector puede consultar [Mitchell, 1996].

2. Algoritmos Genéticos

2.1.¿Que son los algoritmos genéticos?

Basados en modelos computacionales de la evolución biológica natural, los algoritmos genéticos pertenecen a la clase de los algoritmos evolutivos, junto con la programación evolutiva, la evolución de estrategias y la programación genética.

Los **algoritmos genéticos** (AGs) son mecanismos de búsqueda basados en las leyes de la selección natural y de la genética. Combinan la supervivencia de los individuos mejor adaptados junto con operadores de búsqueda genéticos como la mutación y el cruce, de ahí que sean comparables a una búsqueda biológica. Fueron desarrollados por John Holland [Holland, 1975] y Rechemberg [Rechemberg, 1973] que crearon algoritmos de optimización imitando los principios básicos de la naturaleza. Estos algoritmos se utilizan con éxito para gran variedad de problemas que no permiten una solución eficiente a través de la aplicación de técnicas convencionales.

Tiene asimismo aplicaciones variadas en ingeniería, como es el diseño de piezas (turbinas, rotores, etc.), diseño y control de redes, planificación de tareas, síntesis de mecanismos, diseño de tolerancias, etc. y aplicaciones en otros gremios como en sistemas de computación paralelos (paralelización automática de programas secuenciales), química (optimización de procesos de producción, diseño tecnológico y de instalaciones), negocios y comercio (modelización de sistemas económicos complejos, predicción de mercados), medicina (análisis de datos en medicina, diagnóstico automático) o gestión (análisis de datos en gestión, asistentes de gestión, sistemas automáticos de decisión).

Para la ingeniería, los algoritmos genéticos y la programación evolutiva en general presentan oportunidades de plantearse problemas que no permiten una solución eficiente a través de la aplicación de técnicas convencionales.

A continuación, nos centraremos en los algoritmos genéticos, ya que las técnicas empleadas en este proyecto se basan en estos algoritmos evolutivos.

2.2. Antecedentes históricos

Los algoritmos genéticos tienen sus antecedentes en la biología y comienzan con Darwin, que con su libro *El origen de las especies por medio de la selección natural o la preservación de las razas favorecidas en su lucha por la vida* [Darwin, 1859], nos habla sobre los principios de la selección natural.

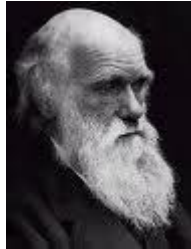


Ilustración 1: Charles Darwin

Los principios básicos de los algoritmos genéticos se derivan de las *Leyes de la Vida Natural* descritos por Darwin [Darwin, 1859]:

- Existe una población de individuos con diferentes propiedades y habilidades. Así mismo existe una limitación sobre el número de individuos que existen en una determinada población.
- La naturaleza crea nuevos individuos con propiedades similares a los individuos existentes.
- Los individuos más prometedores se seleccionan más a menudo para la reproducción de acuerdo con la selección natural.

Los algoritmos genéticos imitan los principios de la vida descritos y los utilizan para propósitos de optimización.

Una de las principales deficiencias del argumento de Darwin es que, a pesar de que la herencia juega un papel preponderante en su teoría, no ofrece una explicación acerca de su funcionamiento. Sin embargo, desde Mendel [Mendel, 1865] se conoce que la herencia se produce a través del código genético presente en las células reproductivas.

2.3.Características

Las características principales de los algoritmos genéticos son:

- Son algoritmos estocásticos. Dos ejecuciones distintas pueden dar dos soluciones distintas.
- Son algoritmos de búsqueda múltiple, luego dan varias soluciones. Realizan la búsqueda mediante toda una generación objetos, no buscan un único elemento.
- Es una búsqueda paramétricamente robusta. Por lo tanto, tiende a converger a medida que se realizan las generaciones.
- No trabajan directamente con los objetos, sino con la codificación de esos objetos, que pueden ser un número, un conjunto de parámetros, etc. A esta codificación la llamamos cromosoma. En este proyecto se representa a cada individuo por un cromosoma definido por una sucesión de números reales, que representan los elementos de la matriz, es decir, se usa representación con números reales.
- Utilizan una función objetivo que da la información de lo adaptados que están los individuos, y no las derivadas u otra información auxiliar.
- Las reglas de transición son probabilísticas, no determinísticas.
- Los mecanismos de los algoritmos genéticos simples son sorprendentemente fáciles, dado que los mecanismos que utiliza son el mecanismo de copia y el de intercambio de partes de las cadenas de cromosomas.

La razón del funcionamiento de este mecanismo es algo más sutil, y el efecto combinado de simplicidad de la operación y la potencia de su efecto son dos de las principales atracciones de la aproximación mediante algoritmos genéticos.

Algunos de los principales atractivos de los algoritmos genéticos como método de optimización son:

- Se pueden utilizar como método de optimización global o semilocal, según interese.
- No hace uso de ningún gradiente, ni conceptos similares. Por lo que se puede usar con señales con mucho ruido, en las que no tienen sentido las ideas asociadas a la derivada. No obstante, si las funciones son derivables, los algoritmos genéticos se pueden usar en la primera fase de minimización para encontrar las cuencas de atracción, y a continuación utilizar, por ejemplo, un método basado en el gradiente.
- Se pueden utilizar en línea.

Los principales inconvenientes son:

- El número de operaciones y por tanto el coste computacional puede ser muy alto si no se toman precauciones en forma de ajuste de la población, migraciones, mutaciones, etc.
- Se puede emplear fácilmente computación en paralelo. Como es lógico, este tipo de método es apropiado cuando se trabaja con funciones que no son suaves, no son convexas y que no se deben aplicar a casos sencillos en que son más rápidos y precisos otros métodos, como los basados en el gradiente.

En el diseño de un algoritmo genético, a la hora de modelar el problema, es muy importante elegir una buena representación de las soluciones posibles y diseñar cuidadosamente la función de evaluación de dichas soluciones. Para acertar en los parámetros del método, se deben asignar correctamente las tasas de probabilidad a los operadores de selección y cruce.

2.4.Representación

Todos los organismos vivos están constituidos por células, y cada célula contiene uno o más *cromosomas* (cadenas de ADN), que le sirven como una especie de “plano” al organismo. Un cromosoma puede ser conceptualmente dividido en *genes* cada uno de los cuales codifica una proteína. En términos generales, se puede decir que un gen se codifica como si fuera un *rasgo*, como puede serlo el color de ojos. Cada gen se encuentra en una posición particular del cromosoma, y está formado por *alelos*.



Ilustración 2: Cromosoma

Se supone que los individuos (posibles soluciones del problema), pueden representarse como un conjunto de parámetros (que denominaremos genes), los cuales agrupados forman una ristra de valores, a menudo referida como cromosoma. Debe existir una representación de estos genes para poder utilizarlos posteriormente en el algoritmo genético y dotarles de unos valores. Se pueden considerar tres tipos básicos de representación o codificación de los genes:

- Binaria: en ella se utiliza un vector cuya longitud es la del número de genes de cada individuo y el valor que puede tomar cada elemento es un número binario.

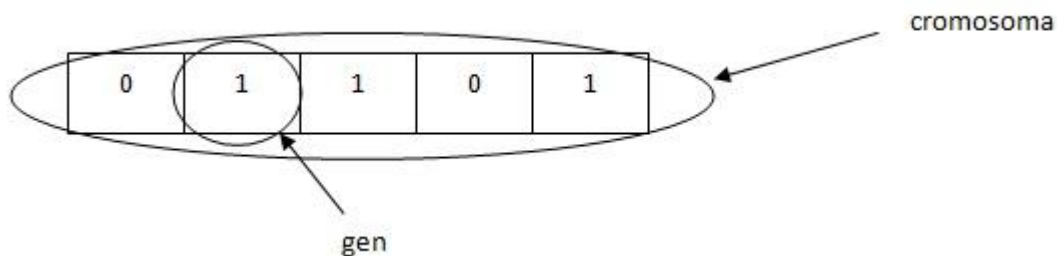


Ilustración 3: Ejemplo de cromosoma de representación binaria

- Entera: en ella se utiliza un vector cuya longitud es la del número de genes de cada individuo y el valor que puede tomar cada elemento es un número entero.

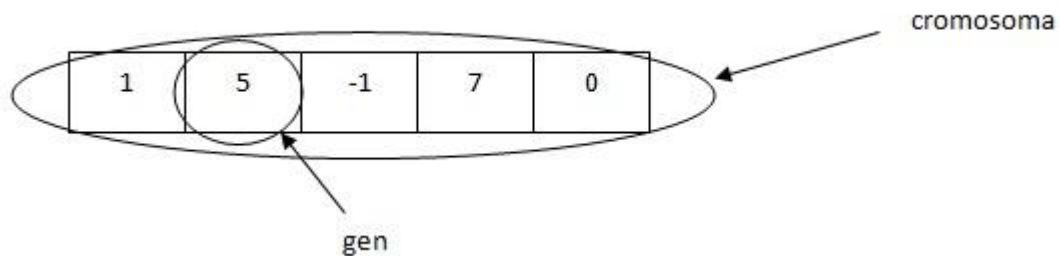


Ilustración 4: Ejemplo de cromosoma de representación entera

- Real: en ella se utiliza un vector cuya longitud es la del número de genes de cada individuo y el valor que puede tomar cada elemento es un número real.

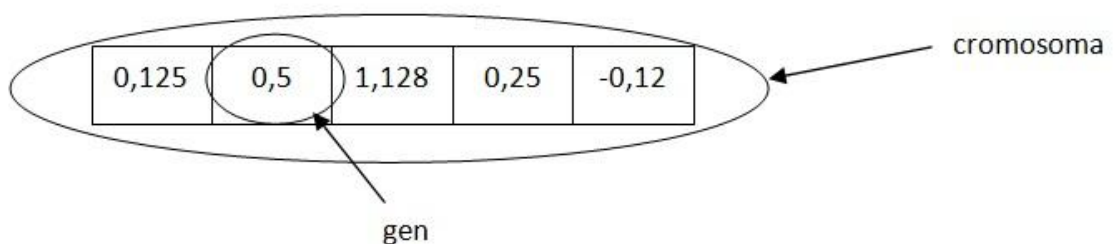


Ilustración 5: Ejemplo de cromosoma de representación real

La representación real de un cromosoma es la que se utiliza en este proyecto.

Un individuo es una solución potencial al problema que se trata. Cada individuo contiene un cromosoma. A un conjunto de individuos se le denomina población. El fitness de un individuo es la evaluación de la función de evaluación e indica qué tan bueno es el individuo (es decir, la solución al problema) con respecto a los demás.

2.5.Algoritmo

Desarrollado por John H. Holland [Holland, 1975], el algoritmo genético opera entonces a nivel de genotipo de las soluciones mediante la siguiente secuencia:

1. Comenzar con una población inicial, la cual puede ser generada de manera aleatoria.
2. Calcular el *fitness* (*aptitud*) de cada individuo.
3. Aplicar el operador de selección con base en el *fitness* de la población.
4. Aplicar los operadores genéticos de reproducción, cruce y mutación a la población actual para generar a la población de la siguiente generación.
5. Ir al paso 2 hasta que la condición de parada se satisfaga.
6. Cuando se cumple la condición de parada, se devuelve al mejor individuo encontrado (bien el mejor de todas las generaciones, bien el mejor de la última generación).

Al igual que en muchas otras heurísticas, el comportamiento del algoritmo genético es altamente dependiente de los parámetros iniciales (tamaño de la población, porcentaje de cruce, porcentaje de mutación, número de generaciones, etc.), por lo que será necesario ajustar esos parámetros para tratar de mejorar la solución para los objetivos del problema.

A cada iteración de este proceso se le denomina una generación. Un algoritmo genético típicamente se itera de 50 a 500 o incluso más generaciones. El conjunto entero de generaciones se denomina una *ejecución*. Al final de una ejecución existen a menudo uno o varios cromosomas altamente adecuados en la población, y que pueden ser elegidos como solución al problema.

La función de evaluación o de fitness de un problema es realmente la función que se desea optimizar. Su diseño es junto con el del genotipo, una de las características más importantes a la hora de encontrar la mejor solución a un problema.

Como se ha podido ver en el algoritmo, en cada generación se selecciona a un conjunto de los mejores individuos (paso 3) y se les modifica para generar la siguiente generación mediante los llamados operadores genéticos (paso 4). Estos operadores son tres: reproducción, cruce y mutación. El operador de cruce intenta simular la reproducción sexual, de tal manera que los individuos resultantes del cruce contendrán información de varios individuos. El operador de mutación simula la mutación biológica, de tal manera que los individuos mutados serán ligeramente diferentes de los individuos originales. El operador de reproducción simplemente copia sin modificación un individuo de una generación a la siguiente. Así en la siguiente generación, algunos individuos simplemente habrán sido copiados, y otros cruzados y/o mutados.

2.6.Función de evaluación y función de aptitud (fitness)

La función de evaluación generalmente es la función objetivo, es decir, es lo que se quiere llegar a optimizar. Es necesario decodificar la solución presente en el cromosoma para evaluarla.

La función de aptitud (fitness) es la que permite valorar la aptitud de los individuos y debe tomar siempre valores positivos.

Ambas funciones suelen ser iguales, pero puede ser que la función objetivo sea muy compleja, tome valores negativos, o no proporcione un valor numérico y, por lo tanto, sea necesario definir una función de aptitud diferente.

En un algoritmo genético la información ha de codificarse para poder trabajar adecuadamente con ella. Cómo hemos visto, existen numerosos sistemas de codificación, aunque nosotros utilizaremos la codificación real.

Una vez definido el sistema de codificación a emplear se verá cómo actúan los operadores básicos de selección, cruce y mutación sobre este código.

2.7. Operadores genéticos

En su forma más simple, un algoritmo genético consta de los siguientes operadores genéticos: *selección*, *reproducción*, *cruce* (crossover) y *mutación*.

2.7.1. Selección

El proceso de selección sirve para escoger a los individuos de la población mejor adaptados, para que actúen de progenitores de la siguiente generación. En la naturaleza existen varios factores que intervienen para que un individuo pueda tener descendencia. El primero de todos es que consiga sobrevivir, ya sea porque no es devorado por depredadores, o porque sea capaz de procurarse alimento. Lo segundo es que encuentre pareja para reproducirse. El último factor es que la combinación de ambos individuos sea apta para crear un nuevo individuo.

Sin embargo, en la realidad es posible que “el mejor” individuo no pueda reproducirse, pero otro individuo de “peor calidad” pueda conseguirlo. Aunque este hecho es menos probable, sigue siendo posible.

En los algoritmos genéticos, la selección es un conjunto de reglas que sirven para elegir a los progenitores de la siguiente generación. Estos progenitores se reproducirán (cruzamiento genético) y generarán descendencia.

Un sistema muy utilizado en los algoritmos genéticos es la selección por torneo (la utilizada para este proyecto). Este sistema consiste en escoger aleatoriamente de la población un cierto número de individuos. De esos individuos se escoge el mejor de todos para ser el padre. Para escoger la madre se repite el proceso: se escoge aleatoriamente a un número de individuos de la población y se elige al individuo con mejor calidad. Este sistema garantiza un mínimo de diversidad, ya que no siempre se elegirá al mejor individuo de la población para tener descendencia. Pero, por el contrario, existen grandes posibilidades de que éste tenga descendencia, ya que si es escogido en algún torneo, será el vencedor.

2.7.2. Reproducción

En este contexto, se entenderá por “reproducción” la clonación de un individuo. Es decir, un individuo pasará a la siguiente generación sin modificación. De esta manera, la reproducción es un operador genético que se contrapone al cruce y la mutación, puesto que estos últimos modifican los individuos que pasan a la siguiente generación. El objetivo de la reproducción es mantener en la siguiente generación a individuos con fitness alta de la presente generación.

Relacionado con el concepto de reproducción está el de “elitismo”, el cual mantiene a los mejores individuos de una generación a la siguiente, para que no se pierda su información.

2.7.3. Cruce

Durante esta fase se cruzan o mezclan los individuos seleccionados en la fase anterior. Es decir, los genes de los dos padres se mezclan entre sí para dar lugar a los diferentes hijos. Existen diversos métodos de cruce, pero los más utilizados son los siguientes:

- **Cruce basado en un punto:** los dos individuos seleccionados para jugar el papel de padres, son recombinados por medio de la selección de un punto de corte, para posteriormente intercambiar las secciones que se encuentran a la derecha de dicho punto. Es decir, los genes del padre1 a la izquierda del punto de corte forman parte del hijo1 y los situados a la derecha formaran parte del hijo2, mientras que con el padre2 sucederá lo contrario.

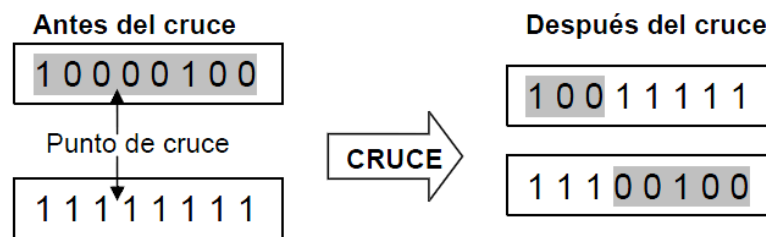


Ilustración 6: Ejemplo de cruce por un punto

- **Cruce punto a punto:** este tipo de cruce es similar al anterior pero realizándose para cada gen de los padres. Por tanto, en este cruce los genes pares del padre1 formarán parte del hijo1 y los genes impares formarán parte del hijo2, mientras que para el padre2 sucederá lo contrario.
- **Cruce multipunto:** en este tipo de cruce se selecciona aleatoriamente la cantidad de puntos que se van a utilizar para el cruce. De esta forma, y de manera análoga al anterior cruce, se irán intercambiando los genes para formar los dos nuevos hijos.

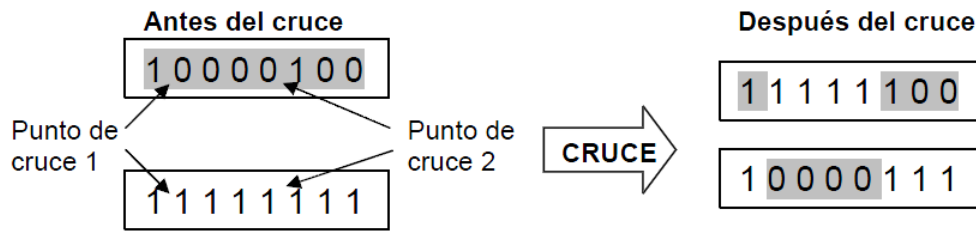


Ilustración 7: Ejemplo de cruce multipunto de dos puntos

- Cruces específicos de codificaciones no binarias: Nuestro proyecto utiliza una codificación de números reales. Para este tipo de codificación se pueden definir, además de los anteriores, otros tipos de operadores de cruce:
 - Media: el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos padres.
 - Media geométrica: cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.
 - Extensión: se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma el valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente.

2.7.4. Mutación

La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en los individuos de la población. Si bien se admite que el operador de cruce es el responsable de efectuar la búsqueda a lo largo del espacio de posibles soluciones, el operador de mutación es el responsable del aumento o reducción del espacio de búsqueda dentro del algoritmo genético y del fomento de la variabilidad genética de los individuos de la población. Existen varios métodos para aplicar la mutación a los individuos de una población, pero el más comúnmente utilizado es el de mutar un porcentaje de los genes totales de la población.

Este porcentaje de genes a mutar se puede seleccionar de dos maneras, de forma fija, especificando el mismo porcentaje de mutación a todas las generaciones del algoritmo genético y de forma variable, es decir, modificando el porcentaje de mutación de una generación a otra, por ejemplo reduciéndolo. De esta manera, se consigue hacer una búsqueda más amplia y global al principio e ir reduciéndola en las siguientes generaciones.



Ilustración 8: Ejemplo de mutación

Con otro tipo de codificaciones (por ejemplo codificación real) existen otras opciones de mutación, aplicadas con una probabilidad generalmente pequeña:

- Mutación al azar: Modifica el valor de un gen asignando con un nuevo valor que se encuentra dentro de un determinado rango. El nuevo valor es independiente del valor previo del gen.
- Mutación gaussiana: Dado un cromosoma p con un gen seleccionado para la mutación i , se le aplica una distribución normal N de media μ_i y desviación estándar σ (parámetro). Alternativamente se puede disminuir el valor de σ a medida que aumenta el número de generaciones.

CAPÍTULO 3: ALGORITMOS DE CLASIFICACIÓN. DISTANCIAS Y PROYECCIONES

•

A lo largo de este capítulo hablaremos de los algoritmos de clasificación, los diferentes tipos que existen y entraremos un poco más en detalle en el algoritmo de clasificación utilizado para la realización de este proyecto.

Además, hablaremos de las distancias y proyecciones, las cuales son utilizadas por gran parte de los algoritmos de este tipo.

3. Algoritmos de Clasificación. Distancias y proyecciones

3.1.Introducción

Uno de los principales objetivos dentro de la Inteligencia Artificial es el de construir sistemas capaces de aprender por sí mismos. El aprendizaje no sólo se encarga de obtener el conocimiento, sino también la forma en que este se representa. El entender también como estas pueden aprender nos puede ayudar a entender las capacidades y limitaciones humanas de aprendizaje.

Existen diversas tareas que se pueden hacer con sistemas de aprendizaje. Entre ellas podemos destacar la segmentación, la estimación, optimización y búsqueda, y la **clasificación**, que es la que nos concierne.

A continuación se definen tres conceptos básicos bajo este contexto:

- Conjunto de datos: Se distinguen dos tipos, el conjunto de entrenamiento y el conjunto de prueba. Para obtener estos, dividimos los datos muestrales en dos partes; una parte se utiliza como conjunto de entrenamiento para determinar los parámetros del clasificador y la otra parte, llamada conjunto de prueba (o test) se utiliza para estimar el error de generalización ya que el objetivo final es que el clasificador consiga un error de generalización pequeño evitando el sobreajuste (o sobre-entrenamiento)
- Modelo: o clasificador, es una conexión entre las variables que son dadas y las que se van a predecir. Usualmente las variables que se van a predecir denominadas variables dependientes y las restantes, variables independientes.
- Aprendiz: es cualquier procedimiento utilizado para construir un modelo a partir del conjunto de datos de entrenamiento. Desde el punto de vista técnico, el aprendizaje se define como el proceso mediante el cual un sistema mejora y adquiere destreza en la ejecución de sus tareas, y tiene la capacidad de poseer inferencia inductiva sobre estas.

Los sistemas de aprendizaje, se pueden encuadrar en cinco tipos:

- Aprendizaje inductivo: Creamos modelos de conceptos a partir de generalizar ejemplos simples. Buscamos patrones comunes que expliquen los ejemplos. Se basa en el razonamiento inductivo: Obtiene conclusiones generales de información específica. El conocimiento obtenido es nuevo. No preserva la verdad (nuevo conocimiento puede invalidar lo obtenido). No tiene una base teórica bien fundamentada.
- Aprendizaje analítico o deductivo: Aplicamos la deducción para obtener descripciones generales a partir de un ejemplo de concepto y su explicación. Se basa en el razonamiento deductivo: Obtiene conocimiento mediante el uso de mecanismos bien establecidos. Este conocimiento no es nuevo (ya está presente implícitamente). Nuevo conocimiento no invalida el ya obtenido. Se fundamenta en la lógica matemática.

- Aprendizaje analógico: Buscamos soluciones a problemas nuevos basándonos en encontrar similitudes con problemas ya conocidos y adaptando sus soluciones.
- Aprendizaje genético: Aplica algoritmos inspirados en la teoría de la evolución para encontrar descripciones generales a conjuntos de ejemplos.
- Aprendizaje conexionista: Busca descripciones generales mediante el uso de la capacidad de adaptación de redes de neuronas artificiales.

Los algoritmos de clasificación se encuadran dentro de los sistemas aprendizaje inductivo.

3.2. Aprendizaje inductivo

El aprendizaje inductivo es la capacidad de obtener nuevos conceptos más generales a partir de ejemplos. Este tipo de aprendizaje conlleva un proceso de generalización/especialización sobre el conjunto de ejemplos de entrada. Los algoritmos implementados son, además, incrementales, es decir, el procesamiento de los ejemplos se realiza uno a uno.

Esta característica, permite visualizar el efecto causado por cada uno de los ejemplos de entrada, en el proceso de obtención del concepto final. Además de la generalización de conceptos, el programa permite clasificar conjuntos de ejemplos a partir de los conceptos obtenidos anteriormente. De este modo, se puede comprobar, para cada ejemplo de un conjunto dado, a que clase pertenece dicho ejemplo.

Este campo lo podemos dividir en dos grandes grupos: los algoritmos supervisados y los no supervisados. Mientras que los algoritmos no supervisados consisten en encontrar la partición más adecuada del conjunto de entrada a partir de similitudes entre sus ejemplos, los algoritmos supervisados intentan extraer aquellas propiedades que permiten discriminar mejor la clase de cada ejemplo, y como consecuencia requieren de una clasificación previa (supervisión) del conjunto de entrenamiento. En este caso, los ejemplos que forman el conjunto de entrenamiento normalmente se componen por pares del estilo <objeto de entrada, clase del objeto>, donde el objeto de entrada suele estar representado por un vector de atributos (o propiedades del objeto). La misión de los algoritmos de aprendizaje automático supervisados es por tanto encontrar el conjunto de atributos que permite predecir con mayor precisión la clase de cada objeto del conjunto de entrenamiento.

3.3. Algoritmos de clasificación por vecindad

Los algoritmos de clasificación por vecindad pertenecen al grupo de las técnicas de clasificación supervisadas, que nos permiten obtener un modelo clasificatorio válido para permitir tratar casos futuros. Los métodos de clasificación exigen la definición de una cierta medida de disimilitud (o distancia) entre los distintos elementos, es decir, precisan de una métrica que ayude a comparar las distancias entre los distintos objetos.

El algoritmo más conocido de este tipo es el algoritmo K-NN o algoritmo de k vecinos más cercanos (k-Nearest Neighbor). K-NN es un algoritmo del tipo *Instance-Based Learning*. Este tipo de algoritmos almacena los datos o instancias utilizadas durante la fase de almacenamiento (conjunto de datos de entrenamiento) para realizar una estimación o clasificación basada en dicho conjunto.

Un objeto es clasificado por la clase más común de sus k vecinos más cercanos. K es un número entero positivo, típicamente pequeño y de valor impar. Si $k=1$, entonces el objeto obtienen la clase del vecino más cercano. Toman los vecinos de un sistema de objetos para los cuales se sabe la clasificación correcta.

Para obtener la distancia a sus vecinos, estos son representados por vectores de posición en un espacio multidimensional, y se usa la función de la distancia euclídea para obtener la distancia hacia ellos, aunque existen variaciones (mas adelante en este capítulo, en el punto de distancias y proyecciones, hablaremos de esta métrica y sus variaciones).

3.3.1. Algoritmo KNN

El algoritmo K-NN funciona de la siguiente manera:

1. Se dispone de un conjunto de datos de entrenamiento o instancias ya clasificadas.
2. Se calcula la distancia de la instancia a clasificar, a las instancias del conjunto de datos de entrenamiento.
3. Se calculará la distancia Euclídea de la nueva instancia respecto a cada una de todas las instancias con las que ya cuenta el algoritmo.
4. Es en este punto, donde entra en funcionamiento el valor k que introduce al algoritmo. La clase resultante de la instancia a clasificar será calculada respecto a las k instancias que minimicen la distancia Euclídea o k vecinos más cercanos. Por ello finalmente la clase será la que más veces se repita en estos k vecinos más cercanos (en caso de igualdad, será aleatoriamente una de ella).

A continuación mostramos un ejemplo de funcionamiento. Se pretende clasificar una nueva instancia mediante el uso de este algoritmo. Se proponen varias situaciones, variando el valor de k .

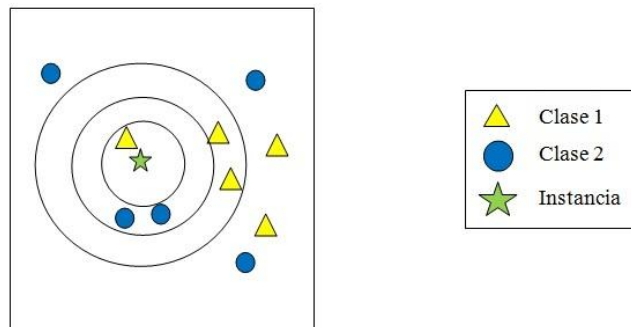


Ilustración 9: Ejemplo algoritmo KNN

Para $k=1$ (círculo más pequeño), la clase de la nueva instancia sería la Clase 1, ya que es la clase de su vecino más cercano.

En cambio para $k=3$, la clase deducida sería Clase 2, ya que entre sus tres mejores vecinos, dos de ellos tendrían dicha clase. Vemos lo importante que es definir un buen valor de k , ya que si $k=5$, volvería a salir ganadora la Clase 1.

La mejor elección de k depende fundamentalmente de los datos; generalmente, valores grandes de k reducen el efecto de ruido en la clasificación, pero crean límites entre clases parecidas. Un buen k puede ser seleccionado mediante una optimización de uso. El caso especial en que la clase es predicha para ser la clase más cercana al ejemplo de entrenamiento (cuando $k=1$) es llamada **Nearest Neighbor Algorithm**, algoritmo del vecino más cercano, y es el caso que se ha utilizado para este proyecto.

3.4. Distancias

Como hemos comentado anteriormente, los algoritmos de clasificación precisan de una métrica que ayude a comparar las distancias entre los distintos objetos. Existen diferentes formas o funciones para determinar la distancia. Lo más común es utilizar la distancia euclídea no ponderada, pero existen variaciones que permiten conseguir una mejor clasificación de los datos.

A continuación vamos a describir en que consiste la distancia euclídea, y veremos dos de sus variaciones, que hemos utilizado durante este proyecto: distancia euclídea ponderada en diagonal y distancia plenamente ponderada.

3.4.1. Distancia euclídea no ponderada

La distancia euclídea es la que normalmente se utiliza en el cálculo de distancias, y por ello es la más conocida y usada. Se define como la longitud del segmento que une dos puntos.

Su función es:

$$d(A, B) \equiv \sqrt{\sum_{i=1}^n (A_i - B_i)^2} = \sqrt{(A - B)^T (A - B)}$$

Ecuación 3: Fórmula de la distancia Euclídea no ponderada

Donde A y B son dos puntos del espacio.

En el siguiente ejemplo se puede observar como para esta función la distancia entre dos puntos es espacio que los separa. Por ello, cualquier punto situado en alguno de los círculos se encuentra a la misma distancia punto central.

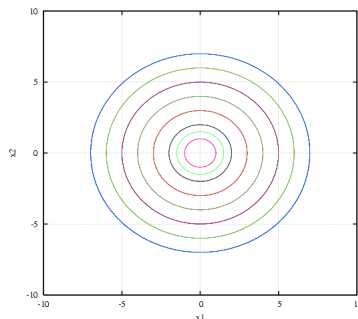


Ilustración 10: Ejemplo de función de distancia euclídea no ponderada

3.4.2. Distancia euclídea ponderada en diagonal

La distancia euclídea ponderada en diagonal es una función de distancia basada en la función de distancia euclídea no ponderada, pero con un factor de ponderación positivo para cada diferencia al cuadrado.

Su función es:

$$d(A, B) \equiv \sqrt{\sum_{i=1}^n (m_i (A_i - B_i))^2} = \sqrt{(A - B)^T M^T M (A - B)}$$

Ecuación 4: Fórmula de la distancia Euclídea ponderada en diagonal

M en este caso, será una matriz diagonal. El efecto que provoca esta función de distancia es similar a escalar cada una de las dimensiones.

En el siguiente ejemplo, se puede ver como esta función de distancia recoge que dos puntos se encuentran a la misma distancia si se encuentran en el mismo óvalo. Por ejemplo, cualquier punto situado en el óvalo verde de la imagen se encuentra más “cerca” del centro que cualquier punto del óvalo azul por ejemplo.

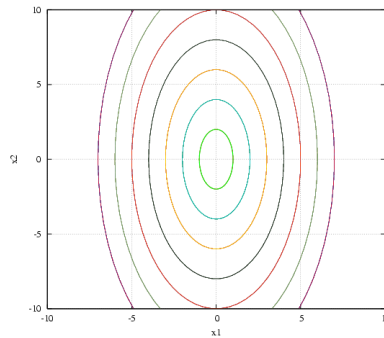


Ilustración 11: Ejemplo de función de distancia euclídea ponderada diagonalmente

3.4.3. Distancia euclídea plenamente ponderada

La distancia euclídea plenamente ponderada es una función que complica un paso más con respecto al caso anterior, la forma de determinar la distancia entre dos puntos.

Su función es:

$$d(A, B) \equiv \sqrt{(A - B)^T M^T M (A - B)}$$

Ecuación 5: Fórmula distancia euclídea plenamente ponderada

En este caso M será una matriz compuesta por cualquier combinación de valores en sus elementos. En el siguiente ejemplo, se puede ver como esta función de distancia recoge que dos puntos se encuentran a la misma distancia si se encuentran en la misma elipse. Por ejemplo, cualquier punto situado en la elipse amarilla de la imagen se encuentra más “cerca” del centro que cualquier punto de la elipse verde por ejemplo.

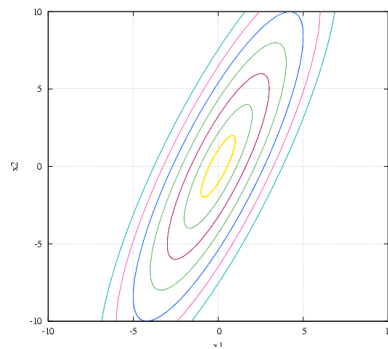


Ilustración 12: Ejemplo de función de distancia euclídea plenamente ponderada

3.5. Proyecciones

La proyección de los datos permite convertir cada dato en un nuevo dato que denominaremos *dato proyectado*. Este nuevo dato obtenido será un dato al cual se le habrán aplicado una serie de factores a cada uno de los atributos o parámetros que componen el dato.

El nuevo dato obtenido, tal y como se ha mencionado con anterioridad, tendrá unos nuevos atributos de acuerdo a la importancia o a los factores que tenga la matriz utilizada.

En el ejemplo que a continuación se presenta, el dato se representa como \mathbf{b} y la matriz que se utilizará para realizar la proyección del dato es \mathbf{a} . El resultado de la proyección será el *dato proyectado* \mathbf{b}' .

$$\begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix} * \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} b'_1 & b'_2 & \dots & b'_n \end{pmatrix}$$

Ilustración 13: Proyección de un dato

No hemos utilizado proyección de datos en este proyecto, por lo que no entraremos más en detalle sobre ello.

CAPÍTULO 4: SISTEMA IMPLEMENTADO

En este capítulo se describe todo el trabajo realizado en el proyecto.

Se comienza el análisis del sistema en el que se describe el objetivo, los requisitos y el funcionamiento global del sistema. Se sigue con el diseño del sistema, en el que se decide cómo llevar a cabo el trabajo descrito en dicho análisis.

A continuación, se realiza una descripción detallada de la implementación del algoritmo genético y el algoritmo de clasificación.

4. Sistema Implementado

4.1. Objetivos y descripción del sistema

4.1.1. Objetivo del proyecto

El principal objetivo de este proyecto es desarrollar un sistema que utilice un algoritmo genético para optimizar funciones de distancia para el algoritmo de clasificación K-NN. Este proceso es equivalente a encontrar la matriz M que define la función de distancia d en la ecuación:

$$d(A, B) \equiv \sqrt{(A - B)^T M^T M (A - B)}$$

Ecuación 6: Distancia euclídea ponderada

La distancia euclídea es equivalente a usar una matriz M igual a la matriz identidad. Se pretende que el algoritmo genético encuentre una matriz M que permita a K-NN encontrar porcentajes de aciertos mejores que la distancia euclídea.

Se van a evolucionar tres tipos de matrices:

- Completa: todos los elementos de la matriz formarán parte del cromosoma del individuo a evolucionar.
- Simétrica: Sólo la diagonal y la parte superior formará parte del cromosoma del individuo a evolucionar.
- Diagonal: Sólo la diagonal de la matriz formará parte del cromosoma del individuo a evolucionar, el resto de los elementos de la matriz serán 0.

Los valores de los elementos de la matriz, se han acotado entre -1 y 1. Esto no debería ser problemático porque si d_M es la distancia correspondiente a la matriz M y d_{kM} es la distancia correspondiente a la matriz M vemos que $d_{kM}(A, B) = k * d_M(A, B)$. El multiplicar todas las distancias por una constante no afecta al algoritmo del vecino más cercano.

A continuación se detalla la configuración del algoritmo genético y detalles de su implementación.

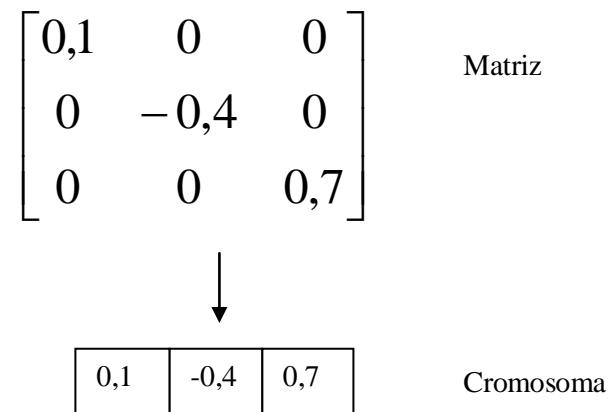
4.1.2. Definición del algoritmo genético

Para la implementación del algoritmo genético, se ha decidido utilizar un framework para Java llamado JGAP (Java Genetic Algorithms Package) [JGAP, <http://jgap.sourceforge.net/>].

Para la representación de los cromosomas, se ha optado por una representación con números reales debido a la naturaleza del problema (matrices con elementos de valores reales). A continuación se muestra un esquema de representación de cada uno de los tipos de matrices como cromosomas para el algoritmo genético.

Matriz diagonal

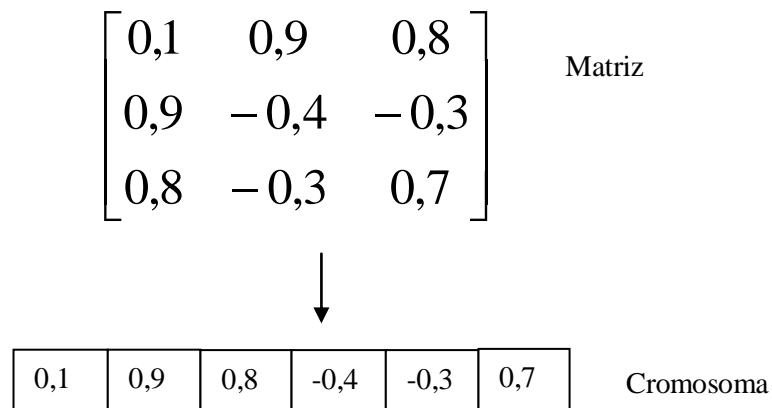
Una matriz diagonal es una matriz cuadrada donde las entradas son todas nulas salvo en la diagonal principal, y éstas pueden ser nulas o no. Su representación como cromosoma ha sido:



Como vemos, solo los elementos de la diagonal forman parte del cromosoma.

Matriz Simétrica

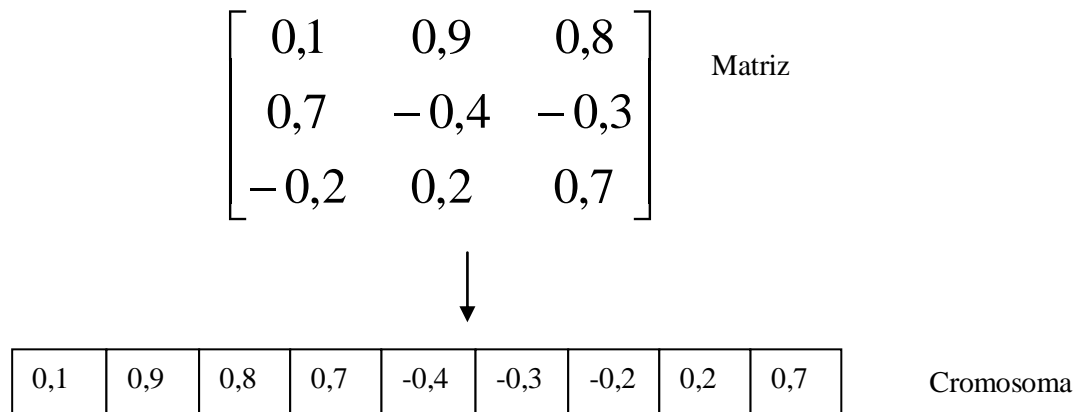
Una matriz simétrica es una matriz cuadrada donde se cumple que $a_{ij} = a_{ji}$ para todo $i, j = 1, 2, 3, 4, \dots, n$. Su representación como cromosoma ha sido:



Como vemos, solo los elementos de la diagonal, y los elementos superiores de la matriz forman parte del cromosoma, ya que los elementos inferiores serán una copia de los superiores. Se recorre la matriz de izquierda-derecha y de arriba-abajo.

Matriz Completa

El termino matriz completa no es estándar, aquí lo utilizamos para saber que se van a evolucionar todos los elementos de la matriz.



Como vemos, todos los elementos de la matriz forman parte del cromosoma. Se recorre la matriz de izquierda-derecha y de arriba-abajo.

4.1.3. Función de Fitness

El algoritmo genético maximiza el valor de la función de Fitness.

La función de Fitness implementada en este sistema es la siguiente:

- Para evaluar el cromosoma A, se transforma en la matriz M según corresponda, dependiendo del tipo de matriz que se está evolucionando.
- Se pasa esta matriz como parámetro a la ejecución del algoritmo de clasificación KNN.
- Se ejecuta el algoritmo de clasificación KNN.
- El Fitness será el número de aciertos que tiene el algoritmo al clasificar los individuos.

4.2. Análisis del sistema

En este apartado vamos a definir los requisitos funcionales y no funcionales del sistema. De cada requisito se especifica:

- Identificador: identifica de forma unívoca cada uno de los requisitos.
- Nombre: resumen del requisito.
- Descripción: explicación con detalle del requisito.

4.2.1. Requisitos no funcionales

Identificador	NF-01
Nombre	Portabilidad
Descripción	El sistema debe poder ejecutarse en el mayor número de sistemas operativos posibles, incluyendo como mínimo a Windows y Unix

Identificador	NF-02
Nombre	Rendimiento
Descripción	Se deberá optimizar dentro de lo posible el coste de ejecución de cada una de las evoluciones, así como la ejecución completa del programa, que finaliza con la obtención de la mejor solución.

Identificador	NF-03
Nombre	Lenguaje de programación
Descripción	Se utilizará como lenguaje de programación Java, utilizando como mínimo la versión de kit de desarrollo JDK 1.5

4.2.2. Requisitos funcionales

Identificador	F-01
Nombre	Configuración
Descripción	Toda configuración del sistema se podrá realizar de forma rápida y a través de un único fichero de configuración. Podrá ser mediante un fichero de texto.

Identificador	F-02
Nombre	Ficheros externos
Descripción	El sistema debe obtener los datos, tanto para los datos de pruebas o entrenamiento y test, de ficheros externos en texto plano. Cada individuo estará en líneas separadas por un retorno de carro. Las características de cada individuo deberán estar separadas mediante comas, tabulaciones o espacios.

Identificador	F-03
Nombre	Fichero único
Descripción	El sistema debe ser capaz de usar el mismo fichero tanto para entrenamiento como para test, pudiendo usar la validación cruzada definiendo el número de particiones si se trata de un único fichero para todo.

Identificador	F-04
Nombre	Número de vecinos
Descripción	El número de vecinos a evaluar utilizado por el algoritmo de búsqueda KNN debe ser configurable.

Identificador	F-05
Nombre	Evolución de la matriz
Descripción	<p>El sistema deberá permitir evolucionar la matriz con la que se optimiza la fórmula del cálculo de la distancia en el algoritmo de búsqueda KNN, de tres formas distintas:</p> <ul style="list-style-type: none"> - Completa: todos los elementos de la matriz formarán parte del cromosoma del individuo a evolucionar. - Simétrica: Sólo la diagonal y la parte superior formará parte del cromosoma del individuo a evolucionar. - Diagonal: Sólo la diagonal de la matriz formará parte del cromosoma del individuo a evolucionar, el resto de los elementos de la matriz serán 0.

Identificador	F-06
Nombre	Número de evoluciones
Descripción	El número total de evoluciones o generaciones del algoritmo genético se debe poder configurar para cada ejecución.

Identificador	F-07
Nombre	Desviación de la mutación
Descripción	La desviación del operador de mutación debe ser configurable en cada ejecución.

Identificador	F-08
Nombre	Tamaño de la población
Descripción	El tamaño de la población total para el algoritmo genético debe ser configurable.

Identificador	F-09
Nombre	Ejecuciones con matriz identidad
Descripción	El sistema debe permitir ejecuciones con la matriz identidad, simulando una ejecución de KNN con distancia euclídea normal.

Identificador	F-10
Nombre	Ficheros de resultados
Descripción	Las ejecuciones deben dejar un sistema de trazas en un archivo para poder realizar consultas posteriores de los resultados.

4.3.Funcionamiento global del sistema

Vamos a intentar detallar el funcionamiento global del sistema. La ejecución será distinta si vamos a usar validación cruzada o no:

- Si vamos a realizar validación cruzada, se usará el mismo fichero para entrenamiento y test, dividiendo los datos del fichero en n particiones (parámetro configurable). Se crearan entonces n hilos de ejecución, cada uno de ellos usando un grupo distinto de $n - 1$ particiones para entrenamiento y 1 partición para test.
- Si no vamos a realizar validación cruzada, se deberá pasar un fichero para entrenamiento y otro fichero para test, realizándose la ejecución sobre el fichero de entrenamiento y el resultado se evaluará contra el fichero de test.

La ejecución también depende de si se va a realizar una evolución de la matriz o no:

- Si no se va a realizar evolución de la matriz, la ejecución lo único que hará es ejecutar el algoritmo de clasificación K-NN usando la matriz identidad, por lo tanto, usando la distancia euclídea normal, y no se usará el algoritmo genético para evolucionar dicha matriz.
- Si se va a evolucionar la matriz (esta es la ejecución más usada), se usará el algoritmo genético para obtener el mejor individuo, pudiéndose evolucionar la matriz de tres maneras distintas:
 - Completa: todos los elementos de la matriz formarán parte del cromosoma del individuo a evolucionar.
 - Simétrica: Sólo la diagonal y la parte superior formará parte del cromosoma del individuo a evolucionar.
 - Diagonal: Sólo la diagonal de la matriz formará parte del cromosoma del individuo a evolucionar, el resto de los elementos de la matriz serán 0.

La mayoría de las ejecuciones se han realizado con validación cruzada, pero se especificará en el capítulo de experimentación la opción elegida en este aspecto.

En el caso de evolucionar la matriz, por cada uno de los dominios se han realizado cuatro ejecuciones distintas:

1. Una con la matriz euclídea, para luego poder comparar si mejora o no el uso del algoritmo genético. La menos costosa de todas ya que no se usa el algoritmo genético.
2. Evolucionando una matriz completa. Esta es la ejecución más costosa.
3. Evolucionando una matriz simétrica. Al reducirse el número de cromosomas del individuo, la ejecución fue menos costosa que en el caso anterior.

4. Evolucionando una matriz diagonal. Al reducirse aun más el número de cromosomas del individuo del algoritmo genético, fue incluso menos costosa que la anterior.

A continuación se muestra el flujo de funcionamiento del sistema en el caso de validación cruzada con evolución de matriz:

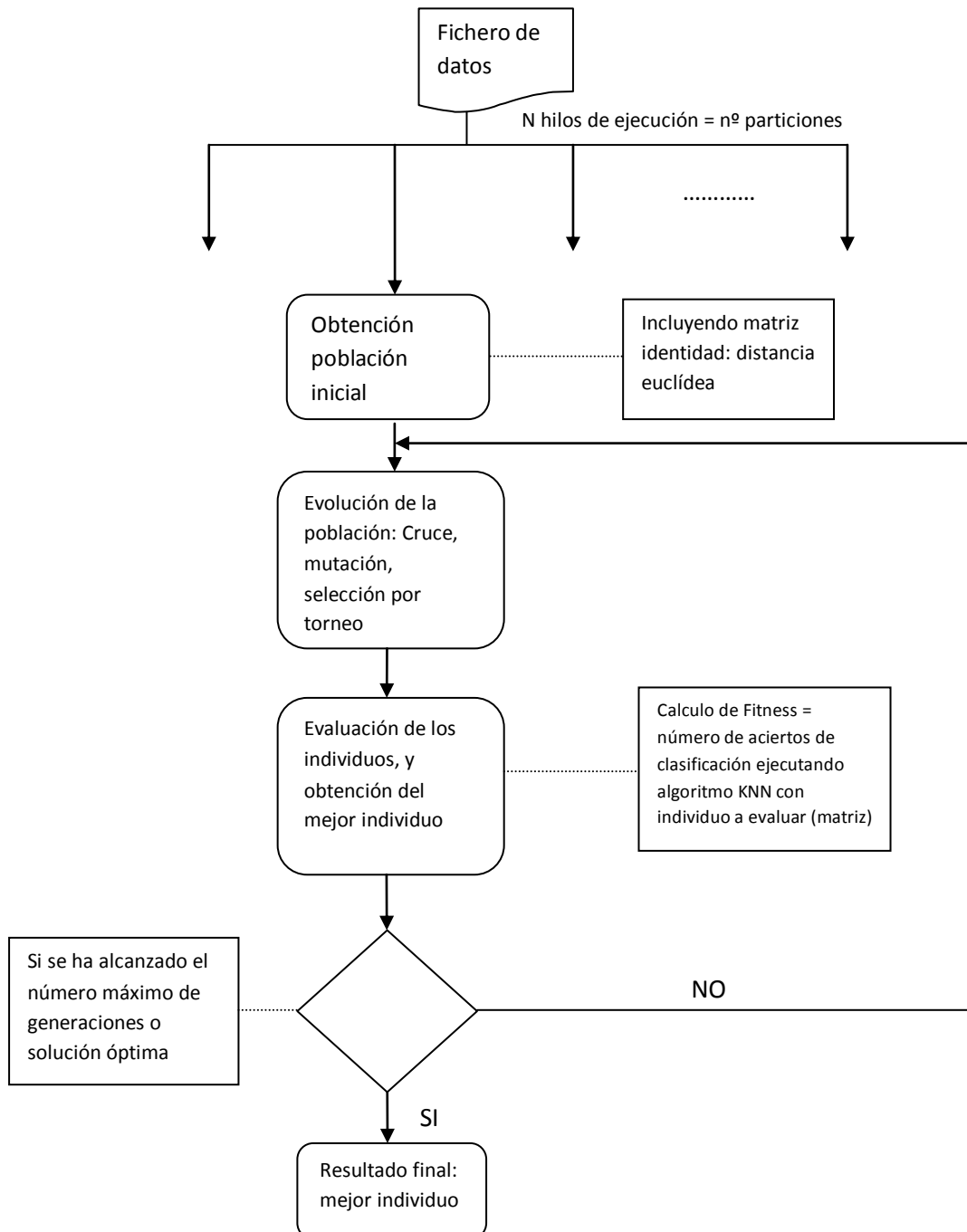


Ilustración 15: Flujo de ejecución

4.4.Diseño del sistema

A continuación se detalla el diseño del sistema. Hemos separado la parte del Algoritmo genético y la parte del algoritmo de búsqueda en distintos paquetes.

A continuación vemos la relación de los paquetes y entraremos en detalle en cada uno de ellos, para finalmente ver un diagrama de clases global.

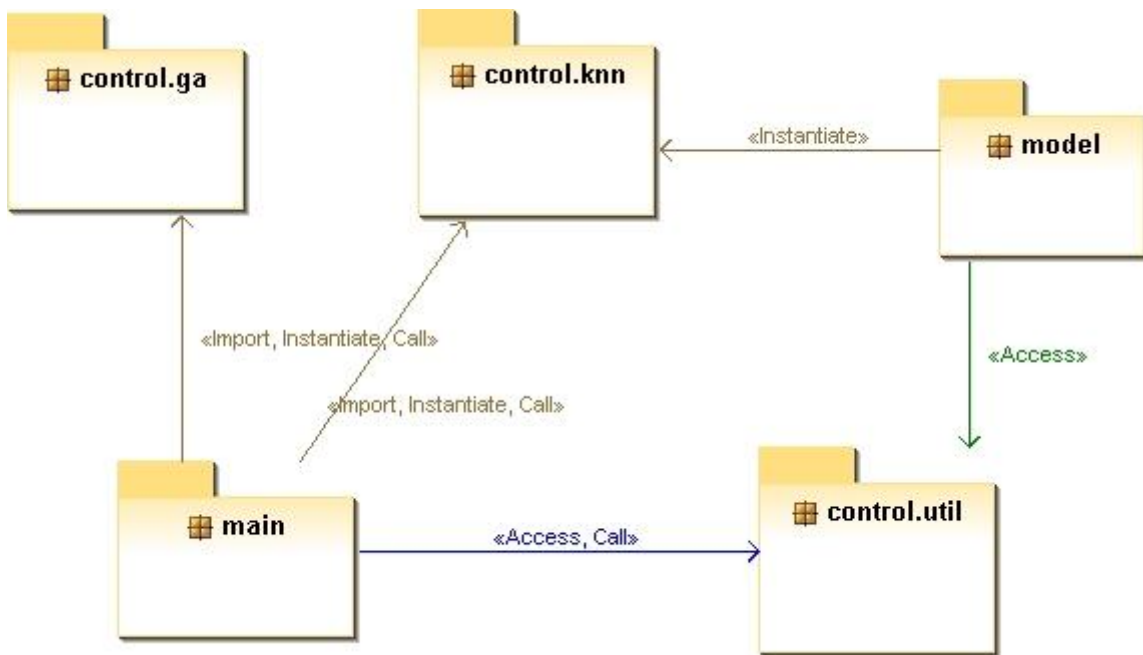


Ilustración 16: Diagrama de clases de paquetes

Como vemos en la ilustración, el paquete principal es “main” el cual se encargara de la ejecución del programa. Vemos la separación entre la codificación del algoritmo genético (paquete “control.ga”) y el algoritmo de búsqueda (“control.knn”). El paquete “model” contiene todas las clases que necesitamos para acceder a los ficheros de los datos, y el paquete “control.util” tiene clases encargadas del manejo de matrices, configuración, etc....

4.4.1. Paquete “main”

A continuación vemos el diagrama de clases del paquete “main”.

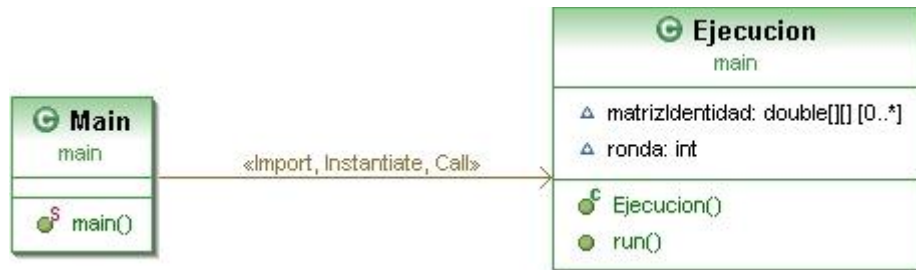


Ilustración 17: Clases del paquete main

Como podemos observar, la clase principal es “Main”. La clase “Ejecucion” hereda de la clase “Thread”. El sistema divide, por cada una de las particiones realizadas del fichero de entrenamiento, la ejecución en hilos individuales, intentando optimizar el rendimiento del sistema.

4.4.2. Paquete “model”

El paquete “model” contiene una única clase, que es la encargada de parsear los ficheros con los datos de entrenamiento y test, según las especificaciones de los requisitos.



Ilustración 18: Clases del paquete model

4.4.3. Paquete “control.util”

A continuación vemos el diagrama de clases del paquete “control.util”.

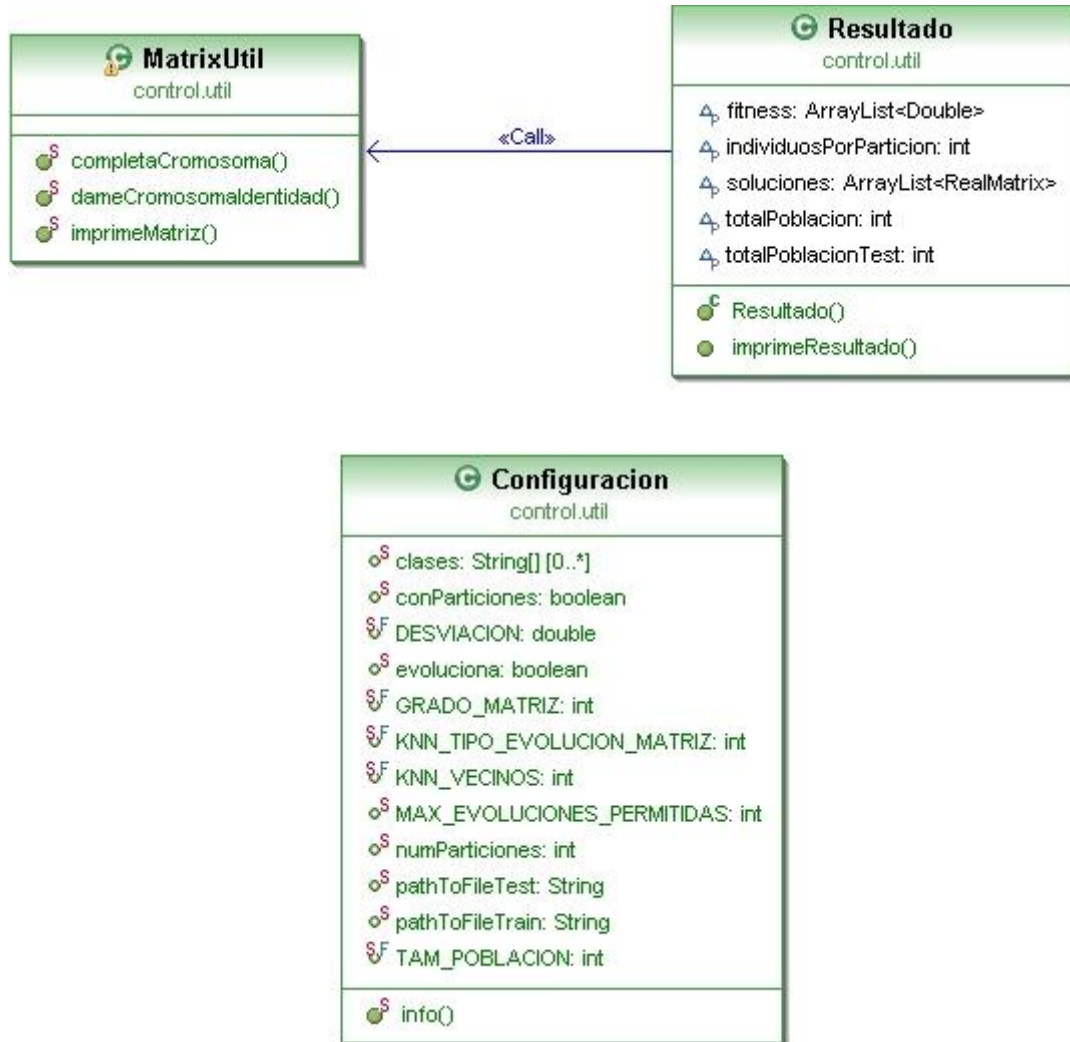


Ilustración 19: Clases del paquete control.util

La clase “Configuracion”, es la encargada de recoger la configuración elegida para la ejecución del sistema. Recoge propiedades como el tamaño de la población, situación de los ficheros de entrenamiento y test, tipo de la evolución del matriz, etc.

La clase “MatrixUtil” contiene funciones útiles para el manejo de matrices, cosa indispensable en el proyecto. Utiliza la clase “RealMatrix” del `commons.math` de apache. Tiene funciones para imprimir la matriz, generar un cromosoma (para el algoritmo genético) a partir de una matriz, etc.

La clase “Resultado” es la encargada de ir almacenando los resultados de cada iteración de ejecución del algoritmo genético (cada generación). Guarda la matriz resultado, el fitness, etc.

4.4.4. Paquete “control.knn”

Este paquete contiene las clases necesarias para implementar el algoritmo de búsqueda Knn. A continuación mostramos el diagrama de clases.

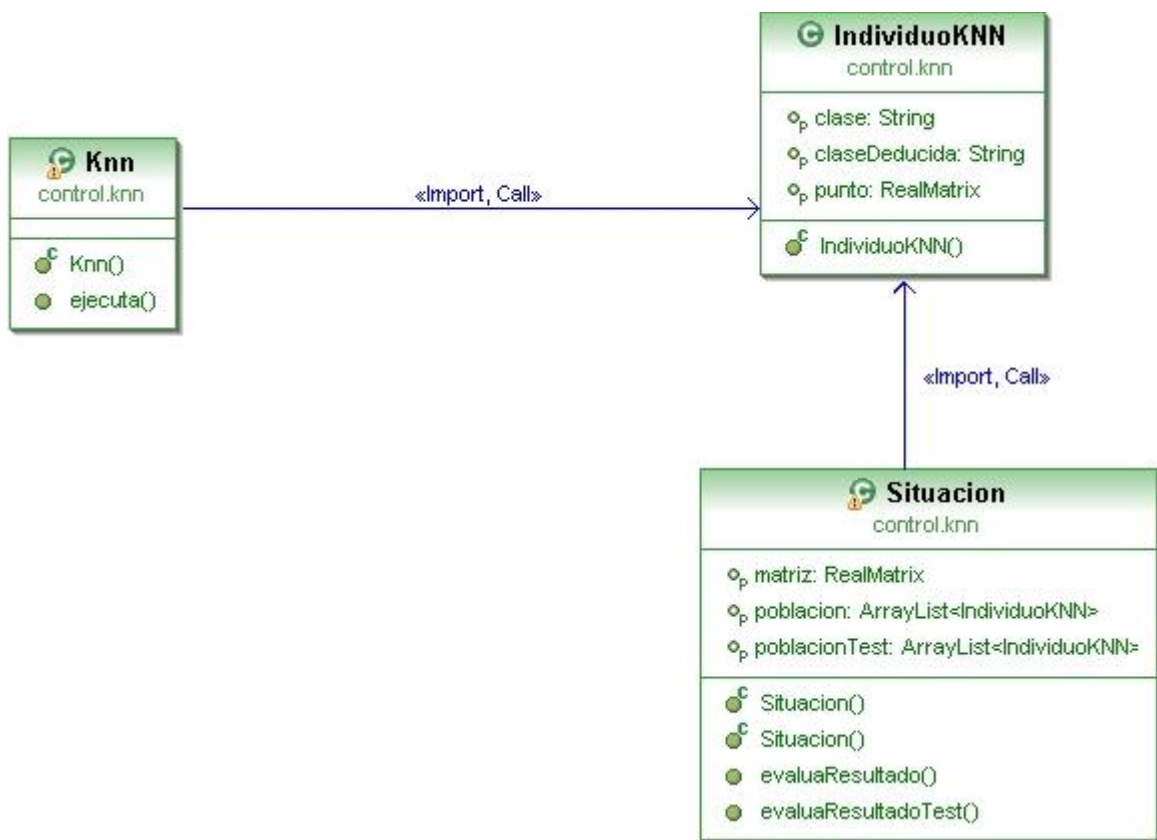


Ilustración 20: Clases del paquete control.knn

La clase Knn ejecuta el algoritmo de búsqueda Knn. Evalúa cada uno de los individuos IndividuoKNN, adjudicándoles la clase obtenida a partir de los k mejores vecinos, que son obtenidos utilizando la formula de distancia euclídea modificada por la matriz candidata M:

$$d(A, B) \equiv \sqrt{(A - B)^T M^T M (A - B)}$$

Ecuación 7: Fórmula de la distancia euclídea ponderada

La clase “Situacion” recoge la situación actual de la ejecución, con todos los individuos de la población de entrenamiento y de test, los resultados obtenidos hasta el momento, y funciones que nos sirven para evaluar el resultado final de la ejecución, comparando la clase deducida con la clase real de cada uno de los individuos, para obtener el número de aciertos que servirá para evaluar la matriz usada como individuo del algoritmo genético.

4.4.5. Paquete “control.ga”

Este paquete contiene las clases que implementan el algoritmo genético. A continuación mostramos el diagrama de clases:



Ilustración 21: Clases del paquete control.ga

La clase **Ga**, es la que implementa el algoritmo genético. Como comentamos anteriormente, se ha utilizado el framework para Java llamado JGAP (Java Genetic Algorithms Package) [JGAP, <http://jgap.sourceforge.net/>].

La clase **FitnessKnn** implementa la función de Fitness que utiliza el algoritmo genético. Lo único que hace es ejecutar el algoritmo de clasificación K-NN, con el individuo que va a ser evaluado (la matriz), y devuelve el número de aciertos que tiene el algoritmo.

CAPÍTULO 5: EXPERIMENTACIÓN

El valor aportado por el proyecto se ha obtenido en la fase de experimentación, en la que se han ido obteniendo resultados con el sistema implementado.

En esta fase del desarrollo del proyecto se han utilizado diferentes dominios de datos para probar el funcionamiento del programa y poder obtener finalmente unos resultados. Los resultados obtenidos nos ayudarán a extraer conclusiones acerca de la combinación del algoritmo de clasificación KNN y de algoritmos genéticos para encontrar la matriz de distancias más adecuada que nos permita obtener las mejores tasas de aciertos en la clasificación de datos desconocidos.

5. Experimentación

5.1. Dominios utilizados en la experimentación

En este capítulo se presentan los resultados obtenidos para una serie de dominios de datos seleccionados. Los resultados obtenidos serán comparados entre para extraer las conclusiones correspondientes.

Como comentamos en el capítulo anterior, el objetivo de la experimentación es encontrar la matriz de distancias que nos devuelva mejores resultados de clasificación para el algoritmo K-NN. Se muestran tanto los resultados de la clasificación usando distancia euclídea, como de las distancias euclídeas generalizadas correspondientes a distintos tipos de matrices que se han evolucionado, que como recordamos son:

- Evolución de una matriz simétrica.
- Evolución de una matriz diagonal.
- Evolución de una matriz completa.

Los dominios utilizados en la fase de experimentación han sido los siguientes:

- Ripley.
- Diabetes.
- Rectas0.
- Rectas45.
- Rectas4D-45g.
- Aleatorio.
- Aleatorio girado.
- EEG.

Para cada uno de los dominios, aparecerá un resumen con información acerca de la configuración utilizada tanto para el algoritmo de clasificación K-NN, como para el algoritmo genético.

También se describirá cada uno de estos dominios, indicando los atributos que componen los datos de los mismos y las clases.

En el Anexo A de este proyecto, se muestran las matrices con las que se obtuvieron todos estos resultados.

5.2. Ripley

5.2.1. Descripción del dominio

El dominio de datos Ripley contiene instancias que han sido generadas artificialmente. Cada una de las instancias cuenta con dos valores reales como coordenadas y una clase que puede 0 o 1. La generación de la clase corresponde a una distribución bimodal que es una composición equilibrada de dos distribuciones. El único cambio referente a las dos distribuciones es centro de ambas.

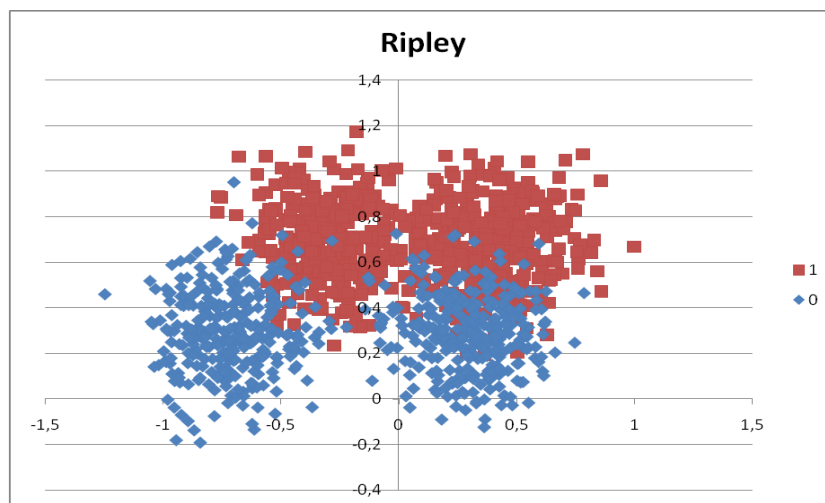


Ilustración 22: Datos del dominio Ripley

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
1000	2	2

Tabla 1: Características del dominio Ripley

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Ripley son:

1. Coordenada x (número real).
2. Coordenada y (número real).
3. Variable de clase :
 - 0

- 1

Se ha realizado validación cruzada, dividiendo el dominio en 5 particiones, por lo tanto se han obtenido cinco resultados, utilizando en cada uno una selección distinta de cuatro particiones para prueba, y una para test.

5.2.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.2.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	89,13%	91,00%	90,00%	91,50%	89,75%	89,50%	90,00%	92,00%
1	90,00%	87,50%	91,50%	83,50%	90,25%	86,50%	91,63%	84,00%
2	90,25%	85,50%	91,38%	85,00%	90,63%	84,50%	91,25%	85,00%
3	90,75%	83,50%	91,63%	81,50%	91,50%	82,00%	91,63%	84,00%
4	87,63%	98,00%	88,63%	98,00%	87,75%	98,00%	88,63%	98,00%
MEDIA	89,55%	89,10%	90,63%	87,90%	89,98%	88,10%	90,63%	88,60%
VARIANZA	0,02%	0,32%	0,02%	0,46%	0,02%	0,38%	0,02%	0,39%

Tabla 2: Resultados del dominio Ripley

Pueden verse las tasas de acierto (en porcentaje) para cada partición, correspondientes a la distancia euclídea y a las diferentes distancias generalizadas que corresponden a las mejores matrices encontradas de los tipos simétrica, diagonal y completa. Se indican

tanto los resultados obtenidos con el conjunto de entrenamiento (columna Prueba) como con el conjunto de test (columna Test).

Como podemos observar, todas las evoluciones de la matriz mejoran los resultados obtenidos en los conjuntos de datos de entrenamiento o prueba, pero no así en los datos de test, cuyo resultados son ligeramente inferiores a los obtenidos utilizando distancia euclídea.

A continuación se muestra una gráfica comparando todos estos resultados:

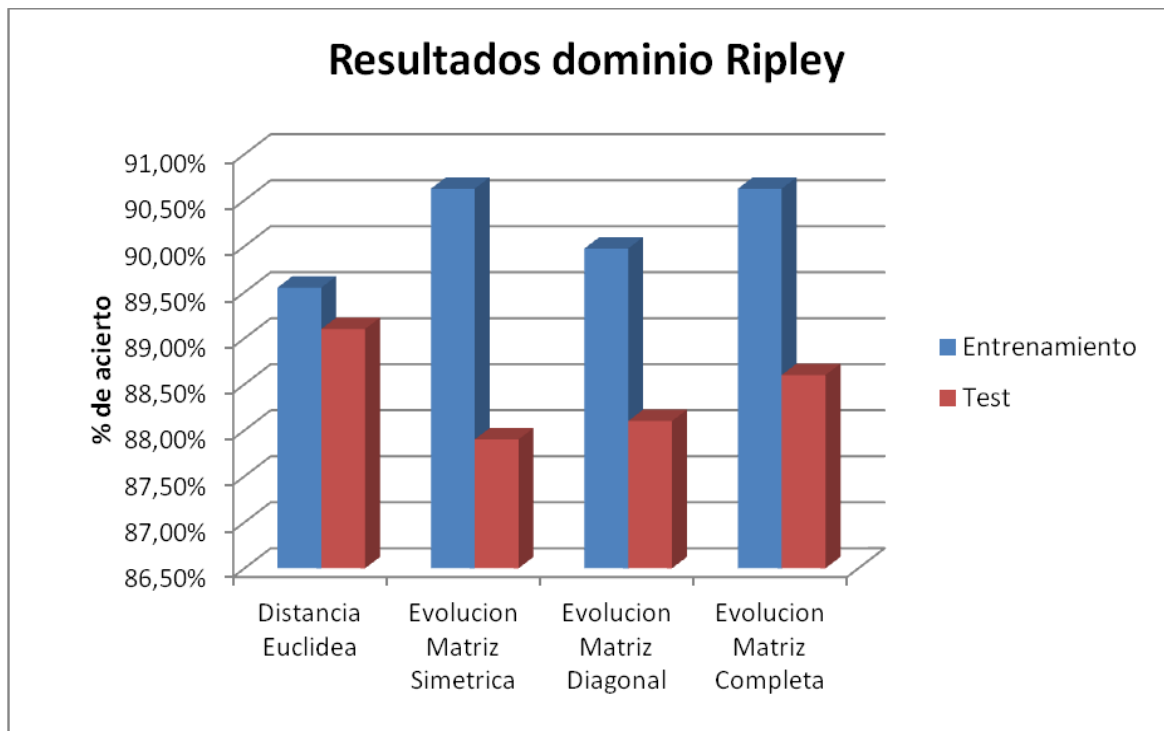


Ilustración 23: Resultados del dominio Ripley

5.3.Diabetes

5.3.1. Descripción del dominio

El dominio Diabetes contiene datos que fueron seleccionados de una base de datos mayor. Concretamente, los datos que se han utilizado corresponden a pacientes que son mujeres con al menos 21 años. Además, todas estas mujeres pertenecen a los Indios PIMA.

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
768	8	2

Tabla 3: Características del dominio Diabetes

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Diabetes son:

1. Número de veces embarazada.
2. Concentración de glucosa en plasma 2 horas en una prueba de tolerancia oral a la glucosa.
3. Presión arterial diastólica (mmHg).
4. Espesor del pliego tricipital (mm).
5. 2 horas de insulina en suero (muU/ml).
6. Índice de masa corporal (kg/m^2).
7. Diabetes función pedigrí.
8. Edad (años).
9. Variable de clase :
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en 5 particiones, por lo tanto se han obtenido cinco resultados, utilizando en cada uno una selección distinta de cuatro particiones para prueba o entrenamiento, y una para test.

5.3.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.3.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	69,77%	62,09%	74,35%	69,28%	75,00%	67,97%	75,82%	67,32%
1	70,75%	60,78%	75,49%	61,44%	76,31%	54,90%	75,82%	53,59%
2	66,99%	62,75%	72,88%	64,05%	73,69%	68,63%	72,22%	63,16%
3	63,56%	75,16%	70,42%	74,51%	71,90%	76,47%	70,42%	61,44%
4	68,95%	66,67%	75,00%	65,36%	74,35%	62,75%	74,67%	66,01%
MEDIA	68,01%	65,49%	73,63%	66,93%	74,25%	66,14%	73,79%	62,30%
VARIANZA	0,08%	0,34%	0,04%	0,26%	0,03%	0,64%	0,06%	0,29%

Tabla 4: Resultados del dominio Diabetes

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos en los conjuntos de datos de entrenamiento o prueba. También se mejoran los resultados con los datos de test, salvo en la evolución de la matriz completa.

A continuación se muestra una gráfica comparando todos estos resultados:

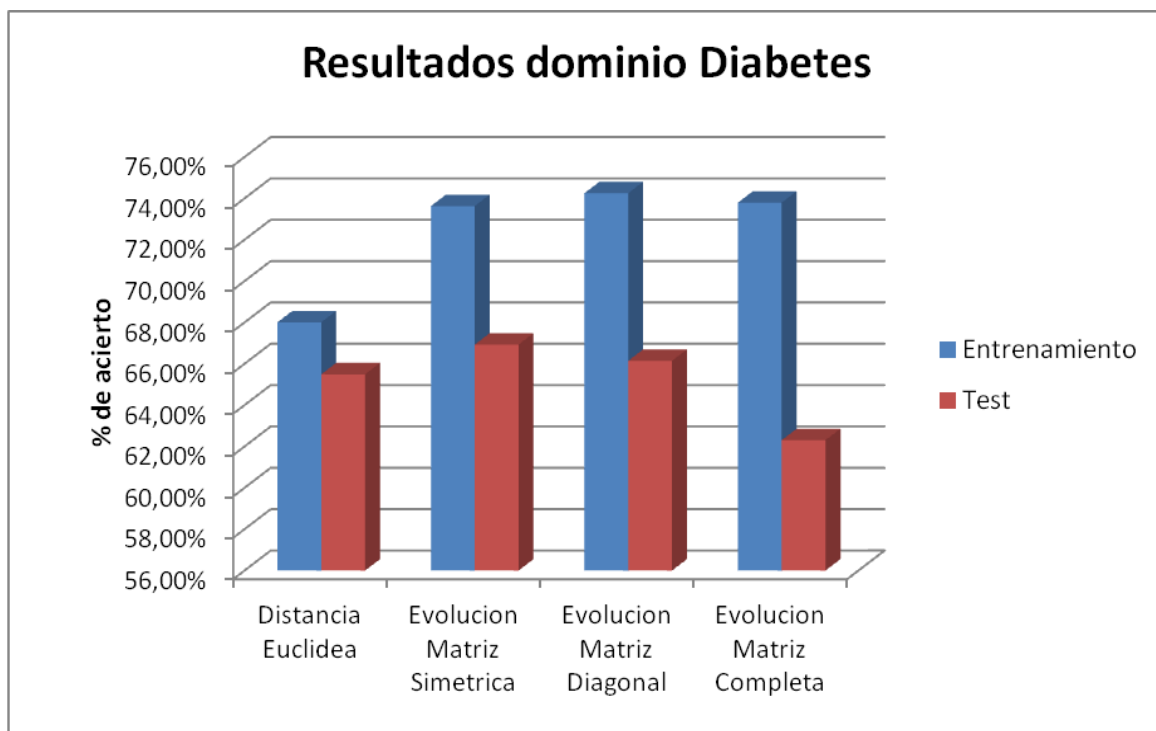


Ilustración 24: Resultados del dominio Diabetes

5.4.Rectas Ruido 0 grados

5.4.1. Descripción del dominio

Dominio con datos generados de forma sintética que representan dos rectas horizontales con ruido.

Los datos de una clase han sido generados a intervalos regulares sobre una recta horizontal que pasa por el punto (0,0). Los datos de la clase contraria se han generado en una recta paralela que pasa por el punto (0,1). También están situados a intervalos regulares de forma que el vecino más cercano a cada punto corresponde al situado en la otra recta, es decir es de la clase contraria. Se ha añadido ruido gaussiano a los puntos situados en ambas rectas.

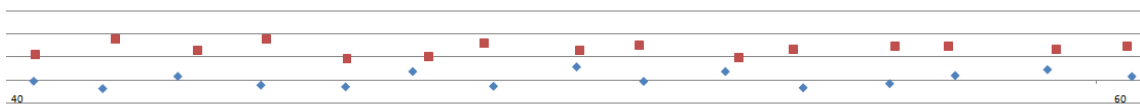
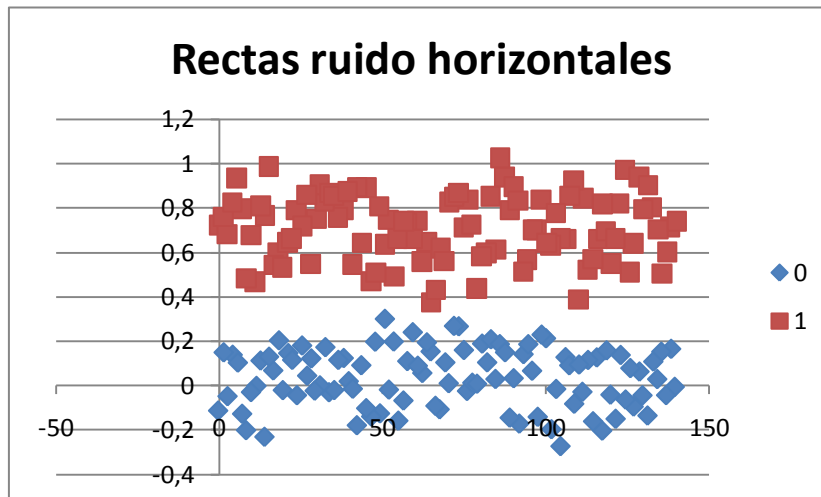


Ilustración 25: Datos del dominio Rectas ruido 0 grados

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
200	2	2

Tabla 5: Características del dominio Rectas ruido 0 grados

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Rectas ruido son:

1. Coordenada X del punto.
2. Coordenada Y del punto.
3. Variable de clase:
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en diez particiones, por lo tanto se han obtenido diez resultados, utilizando en cada uno una selección distinta de nueve particiones para prueba o entrenamiento, y una para test.

Tal como está diseñado el dominio, deberá sacar malos resultados con la distancia euclídea y que bastaría con comprimir uno de los ejes, es decir, una matriz diagonal sería suficiente.

5.4.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.4.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	9,44%	65,00%	100,00%	90,00%	100,00%	95,00%	100,00%	85,00%
1	8,33%	55,00%	99,44%	80,00%	100,00%	100,00%	100,00%	80,00%
2	5,56%	30,00%	100,00%	70,00%	100,00%	70,00%	100,00%	65,00%
3	10,00%	15,00%	100,00%	65,00%	100,00%	80,00%	100,00%	80,00%
4	8,89%	50,00%	100,00%	85,00%	100,00%	90,00%	100,00%	75,00%
5	8,89%	35,00%	100,00%	80,00%	100,00%	95,00%	100,00%	80,00%
6	9,44%	35,00%	100,00%	60,00%	100,00%	60,00%	100,00%	60,00%
7	10,00%	45,00%	100,00%	80,00%	100,00%	95,00%	99,44%	80,00%
8	7,22%	65,00%	100,00%	85,00%	100,00%	95,00%	100,00%	85,00%
9	6,11%	50,00%	100,00%	80,00%	100,00%	80,00%	100,00%	100,00%
MEDIA	8,39%	44,50%	99,94%	77,50%	100,00%	86,00%	99,94%	79,00%
VARIANZA	0,03%	2,52%	0,00%	0,90%	0,00%	1,71%	0,00%	1,21%

Tabla 6: Resultados del dominio Rectas ruido 0 grados

Como podemos observar, se cumplen nuestras expectativas: hay malos resultados con la distancia euclídea y como basta con comprimir uno de los ejes, la matriz diagonal es suficiente para obtener buenos resultados. También se obtienen buenos resultados con las otras matrices.

A continuación se muestra una gráfica comparando todos los resultados:

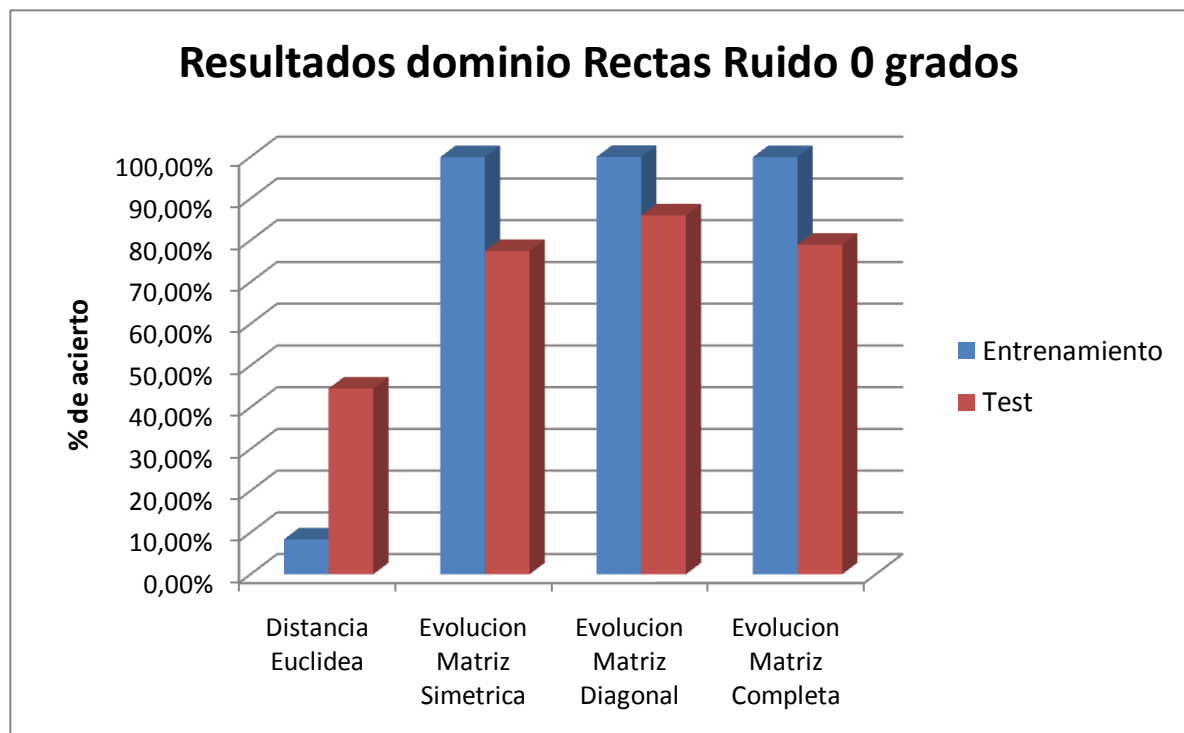


Ilustración 26: Resultados del dominio Rectas ruido 0 grados

5.5.Rectas Ruido 45 grados

5.5.1. Descripción del dominio

Este dominio es similar al dominio anterior, pero los datos han sido girados 45°.

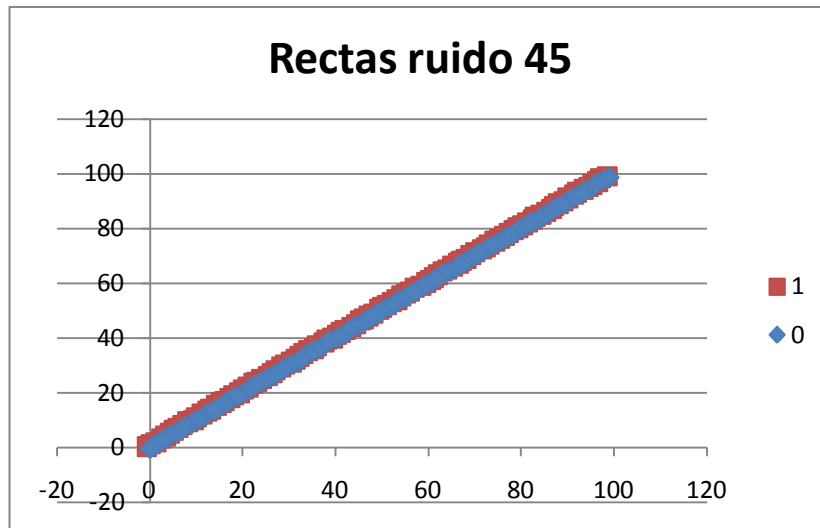


Ilustración 27: Datos del dominio Rectas ruido 45 grados

En esta imagen se puede apreciar como los datos de ambas clases se encuentran muy juntos. Para obtener una mejor visión del conjunto de datos utilizados en este dominio se ha realizado un zoom de una sección de los datos:

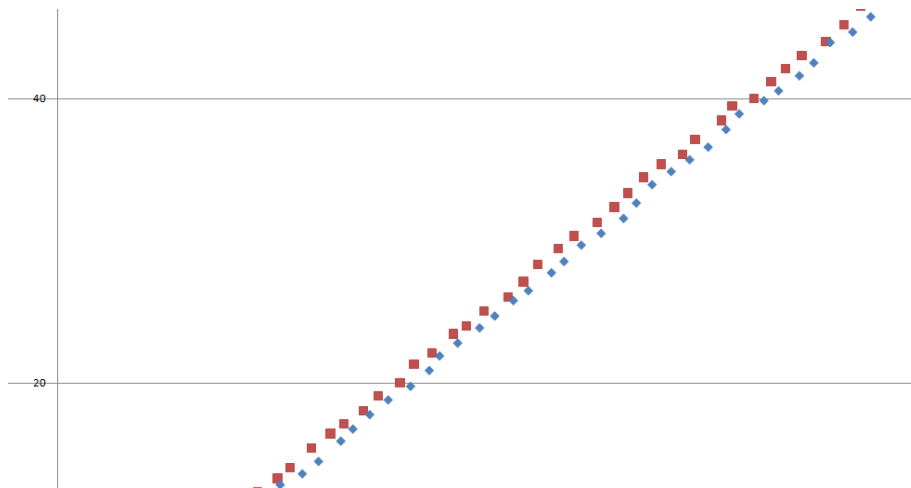


Ilustración 28: Datos del dominio Rectas ruido 45 grados

El zoom realizado muestra como los datos de cualquiera de las clases tienen como único vecino más cercano a un dato de la clase opuesta. Por esta causa, cuando posteriormente

ejecutemos el algoritmo se podrá observar que usando la distancia euclídea el porcentaje de acierto es muy bajo.

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
200	2	2

Tabla 7: Características del dominio Rectas Ruido 45 grados

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Rectas ruido son:

4. Coordenada X del punto.
5. Coordenada Y del punto.
6. Variable de clase :
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en diez particiones, por lo tanto se han obtenido diez resultados, utilizando en cada uno una selección distinta de nueve particiones para prueba o entrenamiento, y una para test.

Decir que ahora no basta con re-escalar los ejes sino que es necesario hacer un giro y un re-escalado, con lo que una matriz diagonal no es suficiente.

5.5.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.5.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	8,89%	35,00%	100,00%	90,00%	10,00%	35,00%	99,44%	95,00%
1	7,78%	35,00%	100,00%	90,00%	9,44%	35,00%	100,00%	35,00%
2	10,00%	50,00%	100,00%	100,00%	10,56%	50,00%	100,00%	100,00%
3	7,78%	45,00%	100,00%	95,00%	8,33%	45,00%	99,44%	55,00%
4	7,22%	55,00%	100,00%	90,00%	10,00%	55,00%	100,00%	95,00%
5	7,78%	55,00%	100,00%	100,00%	8,89%	55,00%	100,00%	100,00%
6	8,33%	60,00%	99,44%	70,00%	13,33%	60,00%	99,44%	70,00%
7	9,44%	50,00%	100,00%	75,00%	10,00%	50,00%	100,00%	75,00%
8	9,44%	45,00%	100,00%	100,00%	10,56%	45,00%	99,44%	85,00%
9	10,00%	70,00%	100,00%	100,00%	11,11%	70,00%	100,00%	100,00%
MEDIA	8,67%	50,00%	99,94%	91,00%	10,22%	50,00%	99,78%	81,00%
VARIANZA	0,01%	1,17%	0,00%	1,16%	0,02%	1,17%	0,00%	4,93%

Tabla 8: Resultados del dominio Rectas ruido 45 grados

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos tanto en los conjuntos de datos de entrenamiento o prueba, como en los de test, pero los resultados con la matriz diagonal no son satisfactorios. Esto se debe a la naturaleza del dominio, donde para encontrar una solución optima se debe realizar un giro en las coordenadas de la recta, cosa que una matriz diagonal es incapaz de realizar.

A continuación se muestra una gráfica comparando todos estos resultados:

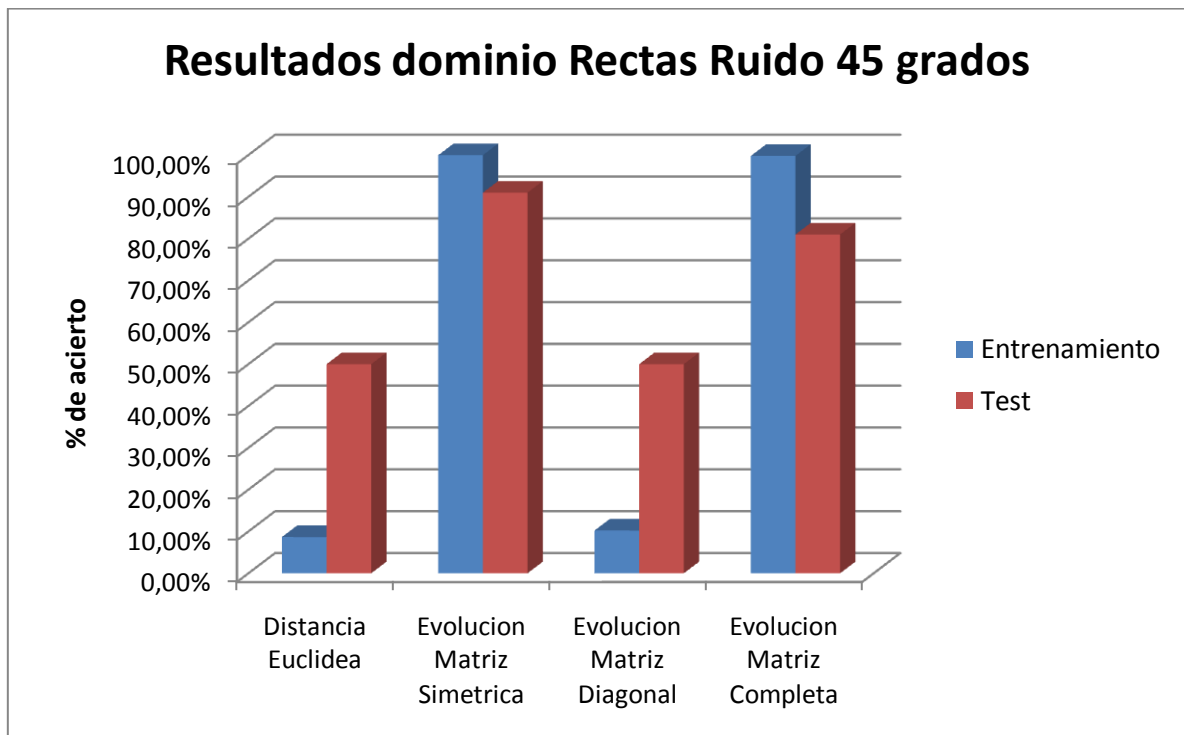


Ilustración 29: Resultados del dominio Rectas ruido 45 grados

5.6.Rectas4D-45g

5.6.1. Descripción del dominio

El dominio Rectas4D contiene datos que representan puntos de rectas en un plano de cuatro dimensiones, formando un ángulo de 45 grados con los ejes de coordenadas. Es similar al dominio de Rectas Ruido 45 grados, pero ahora con cuatro dimensiones.

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
200	4	2

Tabla 9: Características del dominio Rectas4D-45g

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Rectas4D-45g son:

1. Coordenada X del punto.
2. Coordenada Y del punto.
3. Coordenada Z del punto.
4. Coordenada W del punto.
5. Variable de clase :
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en 5 particiones, por lo tanto se han obtenido cinco resultados, utilizando en cada uno una selección distinta de cuatro particiones para prueba o entrenamiento, y una para test.

En este dominio tampoco basta con re-escalar los ejes sino que es necesario hacer un giro y un re-escalado, con lo que una matriz diagonal tampoco debería obtener buenos resultados, igual que en el dominio anterior.

5.6.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación KNN:

- Número de vecinos (K): 1

5.6.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	0,00%	0,00%	100,00%	97,50%	5,63%	7,50%	100,00%	97,50%
1	0,63%	0,00%	100,00%	100,00%	6,25%	5,00%	100,00%	97,50%
2	0,63%	2,50%	100,00%	90,00%	6,88%	2,50%	100,00%	92,50%
3	1,25%	5,00%	100,00%	95,00%	6,88%	7,50%	99,38%	95,00%
4	0,63%	5,00%	100,00%	97,50%	6,25%	5,00%	100,00%	100,00%
MEDIA	0,63%	2,50%	100,00%	96,00%	6,38%	5,50%	99,88%	96,50%
VARIANZA	0,00%	0,06%	0,00%	0,14%	0,00%	0,04%	0,00%	0,08%

Tabla 10: Resultados del dominio Rectas 4D 45 grados

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos tanto en los conjuntos de datos de entrenamiento o prueba, como en los de test, pero los resultados con la matriz diagonal no son satisfactorios. Esto se debe a la naturaleza del dominio, donde para encontrar una solución óptima se debe realizar un giro en las coordenadas de la recta, cosa que una matriz diagonal es incapaz de realizar.

A continuación se muestra una gráfica comparando todos estos resultados:

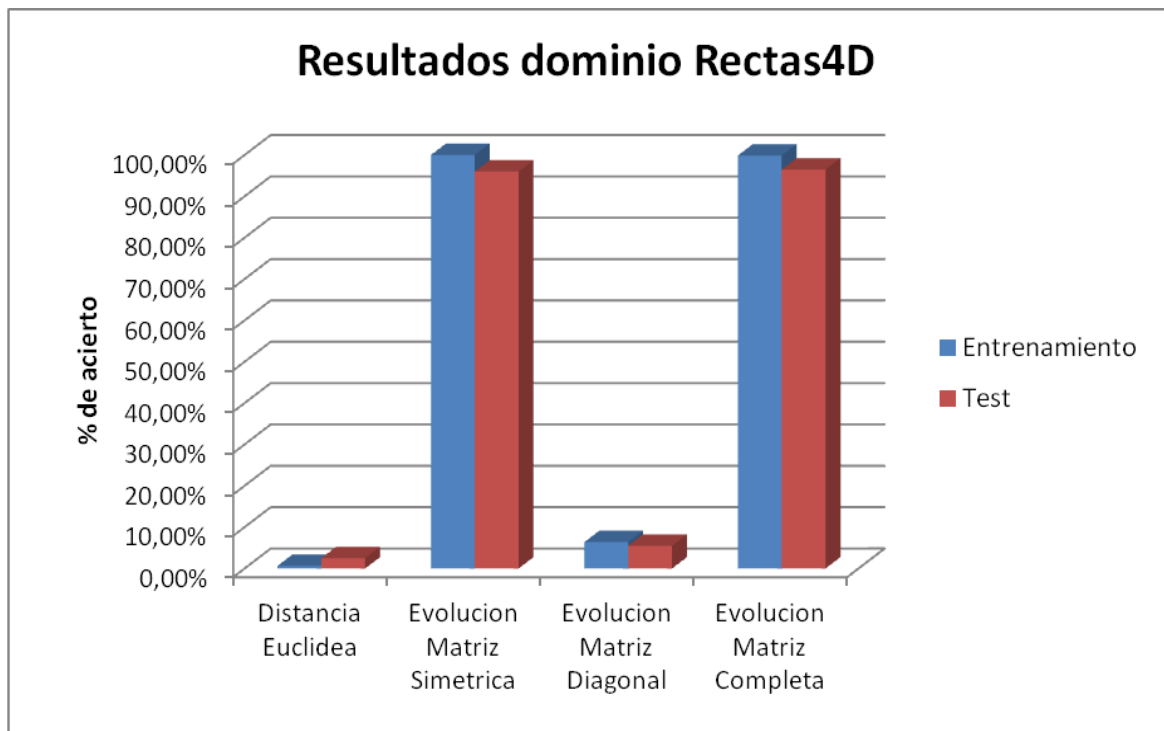


Ilustración 30: Resultados del dominio Rectas 4D 45 grados

5.7. Aleatorio

5.7.1. Descripción del dominio

El dominio aleatorio contiene instancias generadas sintéticamente. Las instancias de este dominio contienen 4 atributos. Dos de ellos representan una recta diagonal, y los otros solamente sirven para generar ruido.

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
300	4	2

Tabla 11: Características del dominio Aleatorio

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Diabetes son:

1. Coordenada 1 (número real).
2. Coordenada 2 (número real).
3. Coordenada 3 (número real).
4. Coordenada 4 (número real).
5. Variable de clase :
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en 5 particiones, por lo tanto se han obtenido cinco resultados, utilizando en cada uno una selección distinta de cuatro particiones para prueba o entrenamiento, y una para test.

Las coordenadas x_3 y x_4 están en el rango de 0..1. Decir que tal como está diseñado, K-NN sacará relativamente malos resultados en la euclídea debido a la influencia de los atributos aleatorios x_3 y x_4 aunque si el rango en estos atributos fuera mayor, su influencia sería mucho más grande.

5.7.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación KNN:

- Número de vecinos (K): 1

5.7.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	91,67%	83,33%	99,17%	93,33%	97,50%	91,67%	99,17%	88,33%
1	91,67%	90,00%	99,17%	96,67%	97,50%	93,33%	99,17%	91,67%
2	92,08%	76,67%	99,17%	93,33%	97,50%	95,00%	99,58%	91,67%
3	90,83%	86,67%	98,33%	91,67%	97,92%	100,00%	99,58%	90,00%
4	89,17%	81,67%	97,92%	90,00%	98,33%	93,33%	98,75%	95,00%
MEDIA	91,08%	83,67%	98,75%	93,00%	97,75%	94,67%	99,25%	91,33%
VARIANZA	0,01%	0,26%	0,00%	0,06%	0,00%	0,10%	0,00%	0,06%

Tabla 12: Resultados del dominio Aleatorio

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos tanto en los conjuntos de datos de entrenamiento o prueba, como en los de test.

Con la matriz diagonal es con la que se obtienen mejores resultados en test, porque basta con anular los atributos aleatorios x3 y x4, y con una matriz de dicho tipo nos basta. Con las otras dos matrices también se consiguen buenos resultados.

A continuación se muestra una gráfica comparando todos estos resultados:

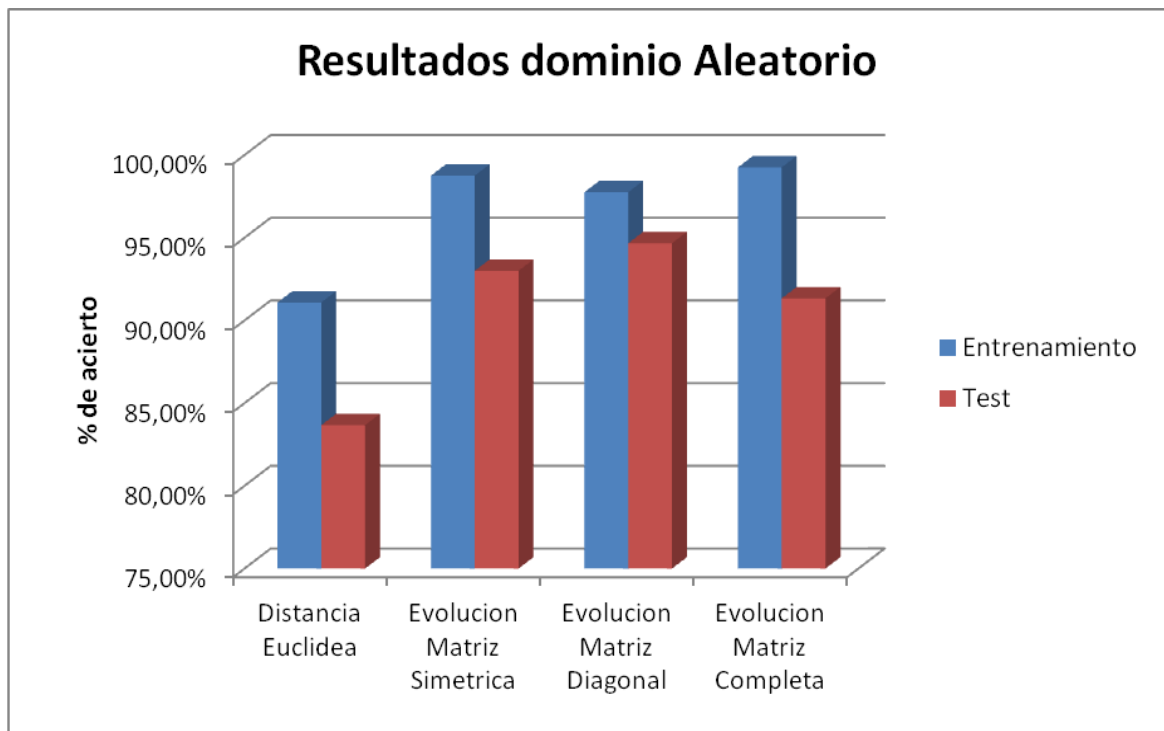


Ilustración 31: Resultados del dominio Aleatorio

5.8. Aleatorio Girado

5.8.1. Descripción del dominio

Dominio con datos generados a partir del dominio Aleatorio (*ver punto anterior*). Los datos de este nuevo dominio han sido generados mediante la multiplicación de los datos del dominio Aleatorio por una matriz 4D. Esta multiplicación provoca una proyección de los datos.

Información acerca de los datos utilizados:

Instancias totales	Atributos	Clases
300	4	2

Tabla 13: Características del dominio Aleatorio Girado

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

Los atributos que componen las instancias del dominio Diabetes son:

6. Coordenada 1 (número real).
7. Coordenada 2 (número real).
8. Coordenada 3 (número real).
9. Coordenada 4 (número real).
10. Variable de clase :
 - 0
 - 1

Se ha realizado validación cruzada, dividiendo el dominio en 5 particiones, por lo tanto se han obtenido cinco resultados, utilizando en cada uno una selección distinta de cuatro particiones para prueba o entrenamiento, y una para test.

5.8.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 50
- Desviación en la mutación: 0.5
- Número de individuos: 80

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.8.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	88,33%	85,00%	95,42%	81,67%	93,33%	83,33%	95,83%	81,67%
1	86,25%	86,67%	95,00%	86,67%	90,42%	86,67%	96,67%	91,67%
2	89,17%	78,33%	95,83%	78,33%	91,67%	81,67%	95,00%	78,33%
3	89,17%	83,33%	97,08%	86,67%	92,50%	83,33%	96,25%	88,33%
4	83,33%	86,67%	95,42%	86,67%	88,75%	80,00%	95,00%	93,33%
MEDIA	87,25%	84,00%	95,75%	84,00%	91,33%	83,00%	95,75%	86,67%
VARIANZA	0,06%	0,12%	0,01%	0,15%	0,03%	0,06%	0,01%	0,42%

Tabla 14: Resultados del dominio Aleatorio Girado

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos tanto en los conjuntos de datos de entrenamiento o prueba, como en los de test, pero los resultados con la matriz diagonal no son satisfactorios en el conjunto de test, que han empeorado respecto a los resultados con la distancia euclídea. Esto es debido a que este tipo de matrices no permite realizar giros.

A continuación se muestra una gráfica comparando todos estos resultados:

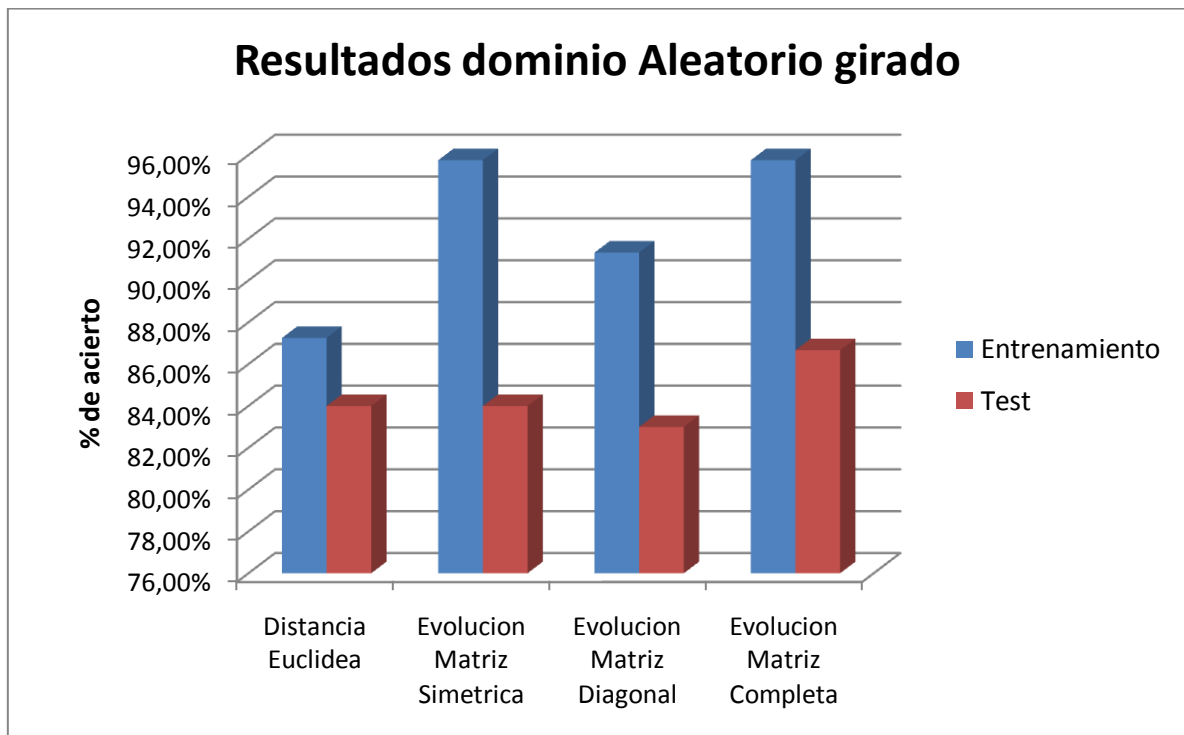


Ilustración 32: Resultados del dominio Aleatorio Girado

5.9.EEG

5.9.1. Descripción del dominio

El dominio de datos EEG contiene instancias basadas en el registro de la actividad bioeléctrica cerebral. Este registro de actividad bioeléctrica cerebral se basa en una exploración neurofisiológica y proviene de la electroencefalografía (EEG). Se pueden tomar datos en diferentes situaciones o condiciones, como por ejemplo: en reposo, durante el sueño, realizando diferentes actividades, etc.

A continuación se muestra una imagen en la que se indica la posición de los electrodos en la cabeza:

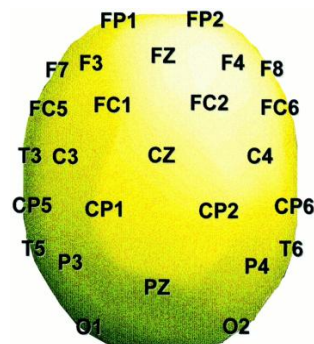


Ilustración 33: Datos del dominio EEG

En esta ocasión, se han utilizado dos conjuntos de datos, uno para entrenamiento o prueba, y otro para test. Además, se disponía de tres conjuntos de pruebas/test distintos, en las que se mezclaban datos de diferentes clases:

- Un primer conjunto con clases 2 y 3

Instancias totales entrenamiento	Atributos	Clases	Instancias totales test
6384	6	2	2064

Tabla 15: Características del dominio EEG con clases 2 y 3

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

1. C3-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-2.
2. C3-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-3.
3. C4-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-2.

4. C4-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-3.
 5. CP1-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-2.
 6. CP1-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-3.
 7. Variable de la clase:
 - 2
 - 3
- Un segundo conjunto con clases 2 y 7

Instancias totales entrenamiento	Atributos	Clases	Instancias totales test
7072	6	2	2480

Tabla 16: Características del dominio EEG con clases 2 y 7

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

1. C3-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-2.
2. C3-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-3.
3. C4-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-2.
4. C4-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-3.
5. CP1-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-2.
6. CP1-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-3.
7. Variable de la clase:
 - 2
 - 7

- Un segundo conjunto con clases 3 y 7

Instancias totales entrenamiento	Atributos	Clases	Instancias totales test
7600	6	2	2464

Tabla 17: Características del dominio EEG con clases 3 y 7

Nota: Los atributos no incluyen el atributo que describe la clase de la instancia.

1. C3-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-2.
2. C3-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C3-3.
3. C4-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-2.
4. C4-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo C4-3.
5. CP1-2. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-2.
6. CP1-3. Actividad eléctrica espontánea en la corteza cerebral medida por el electrodo CP1-3.
7. Variable de la clase:
 - 3
 - 7

Volvemos a destacar que para esta experimentación no se ha realizado validación cruzada ya que los datos utilizados para realizar experimentaciones se encontraban separados en dos conjuntos, uno para entrenamiento y otro para test.

5.9.2. Configuración

La configuración utilizada para el algoritmo genético es la siguiente:

- N° de generaciones: 40
- Desviación en la mutación: 0.5

- Número de individuos: 20

Para el algoritmo de clasificación K-NN:

- Número de vecinos (K): 1

5.9.3. Resultados

En el siguiente cuadro se muestran el resumen con los resultados obtenidos en las experimentaciones:

- Conjunto de datos con clases 2 y 3

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	83,83%	90,02%	84,21%	88,71%	85,26%	90,07%	83,94%	90,02%

Tabla 18: Resultados del dominio EEG con clases 2 y 3

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos en el conjunto de datos de entrenamiento, pero no así en los de test, que sólo mejora la matriz diagonal como en los de test. Los mejores resultados se obtienen con la evolución de la matriz diagonal.

A continuación se muestra una gráfica comparando todos estos resultados:

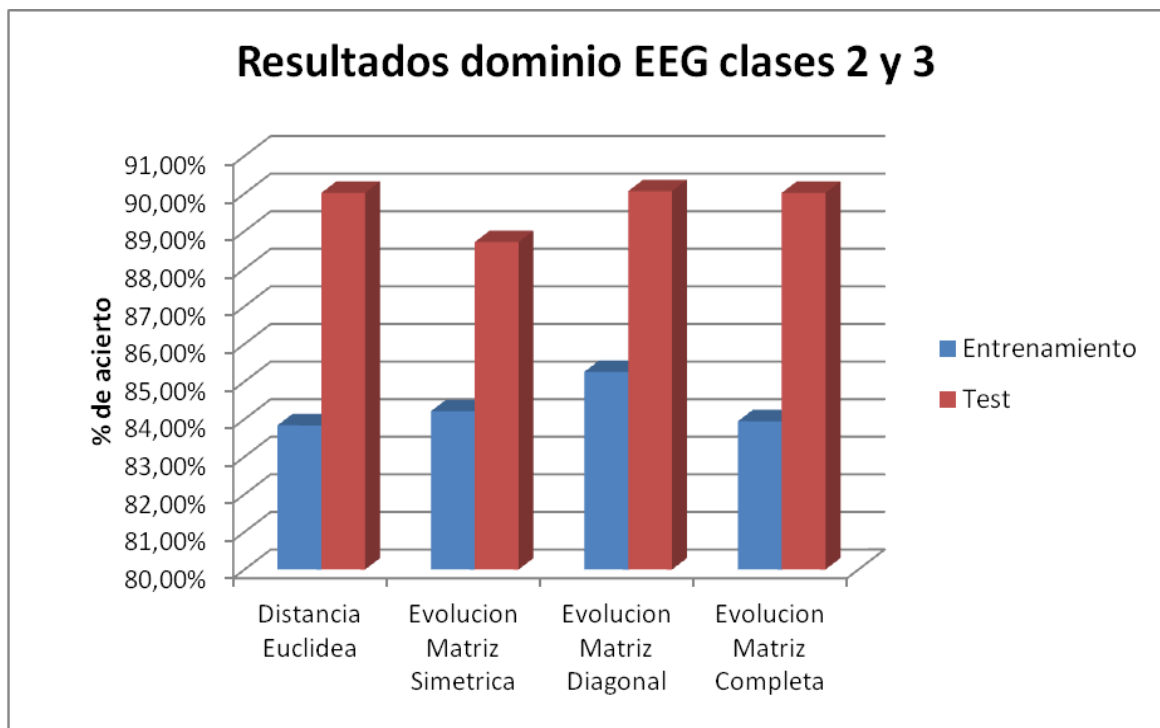


Ilustración 34: Resultados del dominio EEG con clases 2 y 3

- Conjunto de datos con clases 2 y 7

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	87,33%	93,27%	88,32%	93,31%	88,91%	93,23%	87,33%	93,27%

Tabla 19: Resultados del dominio EEG con clases 2 y 7

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos, salvo la matriz completa, que obtiene los mismos resultados. Los mejores resultados se obtienen con la evolución de la matriz diagonal.

A continuación se muestra una gráfica comparando todos estos resultados:

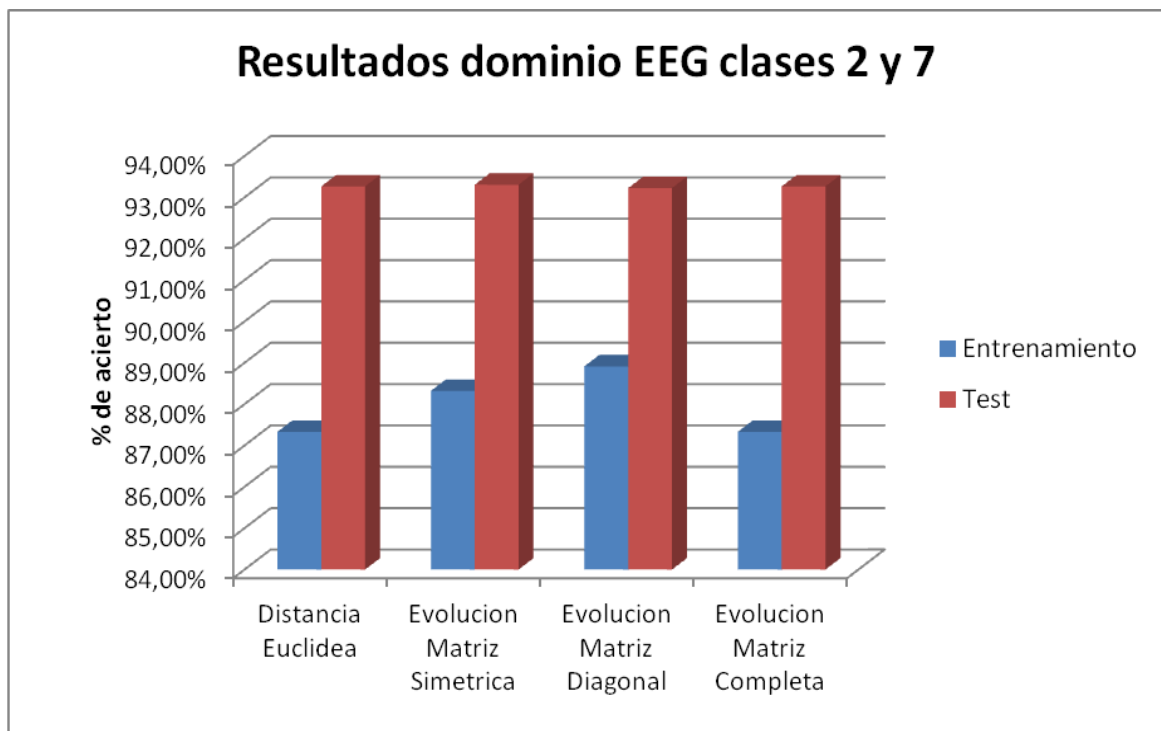


Ilustración 35: Resultados del dominio EEG con clases 2 y 7

- Conjunto de datos con clases 3 y 7

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	82,03%	85,63%	82,99%	88,31%	84,53%	90,22%	82,03%	85,63%

Tabla 20: Resultados del dominio EEG con clases 3 y 7

Como podemos observar todas las evoluciones de la matriz mejoran los resultados obtenidos, salvo la matriz completa, que obtiene los mismos resultados, ya que no ha encontrado durante su ejecución ninguna matriz mejor a la matriz identidad. Los mejores resultados se obtienen con la evolución de la matriz diagonal.

A continuación se muestra una gráfica comparando todos estos resultados:

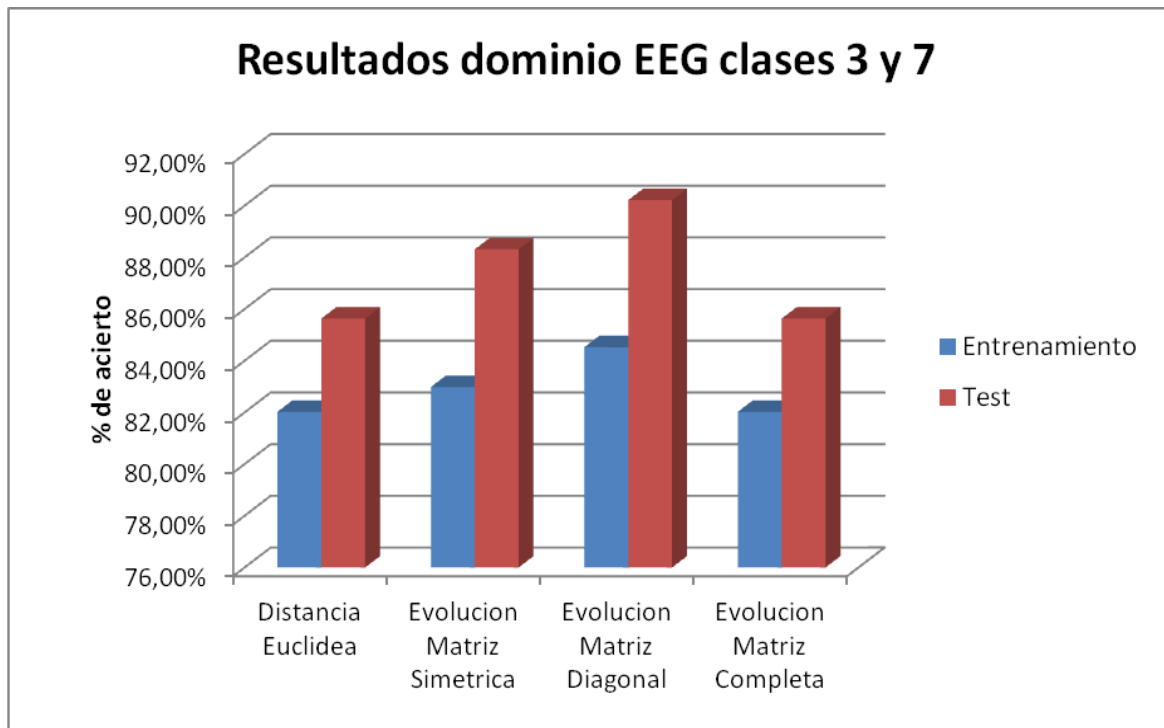


Ilustración 36: Resultados del dominio EEG con clases 3 y 7

CAPÍTULO 6: CONCLUSIONES

Vamos a realizar un pequeño resumen de las conclusiones a las que podemos llegar después de la ejecución del proyecto, comentando brevemente algunas acciones que se podrían llevar a cabo en el futuro.

6. Conclusiones

El principal objetivo de este proyecto era diseñar y desarrollar un sistema que utilizara un algoritmo genético para optimizar funciones de distancia para el algoritmo de clasificación K-NN. Podemos decir que este proceso se ha llevado a cabo de forma satisfactoria, y que los resultados han sido los esperados.

En casi la totalidad de las experimentaciones, el uso de las funciones de distancia evolucionadas para el algoritmo clasificación K-NN mediante el algoritmo genético han mejorado los resultados obtenidos usando la función de distancia euclídea. A continuación mostramos un resumen de los resultados obtenidos:

	Distancia Euclídea	Evolución Matriz Simétrica	Evolución Matriz Diagonal	Evolución Matriz Completa
	Test	Test	Test	Test
RIPLEY	89,10%	87,90%	88,10%	88,60%
DIABETES	65,49%	66,93%	66,14%	62,30%
RECTAS 0	44,50%	77,50%	86,00%	79,00%
RECTAS 45	50,00%	91,00%	50,00%	81,00%
RECTAS 4D	2,50%	96,00%	5,50%	96,50%
ALEATORIO	83,67%	93,00%	94,67%	91,33%
ALEATORIO GIRADO	84,00%	84,00%	83,00%	86,67%
EEG23	90,02%	88,71%	90,07%	90,02%
EEG27	93,27%	93,31%	93,23%	93,27%
EEG37	85,63%	88,31%	90,22%	85,63%
	68,82%	86,67%	74,69%	85,43%

Tabla 21: Resumen de resultados

En casi la totalidad de las experimentaciones, se han encontrado matrices que mejoraban los resultados de K-NN, salvo en aquellos dominios donde la clasificación de la matriz euclídea era ya bastante buena. Esto es debido a un sobre aprendizaje de las matrices, ya que en los conjuntos de entrenamiento todas han mejorado, o como poco, obtenían los mismos resultados.

Como cabía esperar, observando el porcentaje medio de acierto, las matrices simétricas o completas parecen que dan mejores resultados que la matrices diagonales. Pero esto no es del todo cierto, ya que si no consideramos los dominios en los que obtener una buena matriz diagonal era imposible por la característica de estos, como son RECTAS 45 o RECTAS 4D, los resultados se igualan bastante, por lo que se podría proponer como un trabajo futuro comprobar este hecho. Personalmente creo que es porque, al tratarse de un individuo genético más pequeño que las otras dos matrices, a la hora de evolucionar mediante el algoritmo genético parece que lo hace de mejor manera. Aumentando el número de ejecuciones de las otras matrices, tal vez se llegarían a

obtener mejores resultados, pero como he comentado, sería un buen trabajo para realizar en el futuro.

También la lógica nos dice que las matrices completas deberían ser mejores que las simétricas, pero esto ocurre sólo en el 50% de los casos. Puede ser debido a que el número de parámetros extra conlleva un sobre aprendizaje o que requieran de más ciclos genéticos para ser optimizados.

Evolucionar la matriz simétrica parece ser suficiente, ya que en general, es la que mejores resultados ha obtenido.

ANEXO A: MATRICES RESULTADO

Este anexo contiene las matrices resultado (la que mejor resultado obtuvo en la ejecución del algoritmo genético) de cada una de las ejecuciones de la experimentación.

Se volverá a mostrar el resumen de los resultados obtenidos y veremos las matrices con las que se obtuvieron dichos resultados.

ANEXO A: Matrices Resultado

1. Matrices resultado del dominio Ripley

Recordamos los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	89,13%	91,00%	90,00%	91,50%	89,75%	89,50%	90,00%	92,00%
1	90,00%	87,50%	91,50%	83,50%	90,25%	86,50%	91,63%	84,00%
2	90,25%	85,50%	91,38%	85,00%	90,63%	84,50%	91,25%	85,00%
3	90,75%	83,50%	91,63%	81,50%	91,50%	82,00%	91,63%	84,00%
4	87,63%	98,00%	88,63%	98,00%	87,75%	98,00%	88,63%	98,00%
MEDIA	89,55%	89,10%	90,63%	87,90%	89,98%	88,10%	90,63%	88,60%
VARIANZA	0,02%	0,32%	0,02%	0,46%	0,02%	0,38%	0,02%	0,39%

Tabla 22: Resultados del dominio Ripley

1.1. Matriz Simétrica

Partición 0

$$\begin{bmatrix} 0.23967287466525766 & 0.9184313246649828 \\ 0.9184313246649828 & 0.26076320549429044 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.6801571642886479 & -0.26267356231123795 \\ -0.26267356231123795 & -0.23563737452784617 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.4503860838777751 & -0.07938986566023898 \\ -0.07938986566023898 & -0.9147593088339876 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.8035051612270718 & 0.16699647072384438 \\ 0.16699647072384438 & 0.6202402001313341 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.22707554737732294 & 0.9561779096899587 \\ 0.9561779096899587 & 0.34532298112389403 \end{bmatrix}$$

1.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} 0.14392628945633423 & 0.0 \\ 0.0 & 0.8720131348155529 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.35971333227507385 & 0.0 \\ 0.0 & 0.9014511412513533 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.136184158309977 & 0.0 \\ 0.0 & -0.29674346037459276 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.7077448329390896 & 0.0 \\ 0.0 & -0.43306580608770684 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.19524025153003155 & 0.0 \\ 0.0 & -0.16973908257864756 \end{bmatrix}$$

1.3. Matriz Completa

Partición 0

$$\begin{bmatrix} 0.25742143607550294 & -0.38941322759457586 \\ 0.48060677215639225 & 0.025227507047212994 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.02003519969013645 & 0.9870151969735064 \\ 0.17713997592138864 & 0.4450486075315139 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.5493598687165998 & 0.2206286189037674 \\ -0.7006658057206012 & -0.8889281298596654 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.5995878274292064 & -0.4234222183035936 \\ 0.7193547987084412 & 0.3055526476296202 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.5910885669128404 & -0.32147760156449146 \\ 0.6076784048371688 & 0.3702557640390247 \end{bmatrix}$$

2. Matrices resultado del dominio Diabetes

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	69,77%	62,09%	74,35%	69,28%	75,00%	67,97%	75,82%	67,32%
1	70,75%	60,78%	75,49%	61,44%	76,31%	54,90%	75,82%	53,59%
2	66,99%	62,75%	72,88%	64,05%	73,69%	68,63%	72,22%	63,16%
3	63,56%	75,16%	70,42%	74,51%	71,90%	76,47%	70,42%	61,44%
4	68,95%	66,67%	75,00%	65,36%	74,35%	62,75%	74,67%	66,01%
MEDIA	68,01%	65,49%	73,63%	66,93%	74,25%	66,14%	73,79%	62,30%
VARIANZA	0,08%	0,34%	0,04%	0,26%	0,03%	0,64%	0,06%	0,29%

Tabla 23: Resultados del dominio Diabetes

2.1. Matriz Simétrica

Partición 0

-0.5835439548826218	-0.7820232835691721	-0.6006519453441441	-0.3240685027941741	0.43950319687248307	-0.5166920251239513	0.5692691265806066	0.7963080416146274
-0.7820232835691721	0.6229958668629075	0.6603939097560627	-0.2725394588097676	0.33837391690986074	0.9521810206456167	0.05664272329175035	-0.4962523405759759
-0.6006519453441441	0.6603939097560627	-0.33244561106521586	-0.8950887313569449	0.356485183078758	0.23329799728562461	-0.49572550017821093	0.08728663125100011
-0.3240685027941741	-0.2725394588097676	-0.8950887313569449	0.8697874378360191	0.44887269618493	0.1622129275794505	-0.6237700965827699	0.850773331919699
0.43950319687248307	0.33837391690986074	0.356485183078758	0.44887269618493	0.6762801922525523	0.6478025692041762	0.820244896593951	0.7665213323565945
-0.5166920251239513	0.9521810206456167	0.23329799728562461	0.1622129275794505	0.6478025692041762	0.6574743474958765	0.974759201213649	0.9991746625831803
0.5692691265806066	0.05664272329175035	-0.49572550017821093	-0.6237700965827699	0.820244896593951	0.974759201213649	0.39995239006546557	0.970944501563805
0.7963080416146274	-0.4962523405759759	0.08728663125100011	0.850773331919699	0.7665213323565945	0.9991746625831803	0.970944501563805	0.5774822095235685

Partición 1

0.6706998066190809	-0.4189436422350695	0.7401862208272911	-0.5325589295431442	0.8624959834449527	0.8316170856527341	0.9113693114122873	-0.6774517497022479
-0.4189436422350695	0.6514225418442335	0.38913090889445834	0.07790518629049537	0.6010092294921031	-0.5612702940660266	0.304786380165492	0.528905447612025
0.7401862208272911	0.38913090889445834	0.036253973583450616	-0.35972436962600174	-0.46449741703185954	-0.12903221276889332	-0.26404953786921126	0.9209664629352328
-0.5325589295431442	0.07790518629049537	-0.35972436962600174	-0.34546117102782203	0.13126525753399076	-0.3908045103195993	-0.9198093930452853	-0.634989016960501
0.8624959834449527	0.6010092294921031	-0.46449741703185954	0.13126525753399076	0.2163889149184206	0.18665020650257358	0.18665020650257358	0.48246891608780385
0.8316170856527341	-0.5612702940660266	-0.12903221276889332	-0.3908045103195993	0.18665020650257358	0.11068740518772924	0.11371722914850879	-0.29722100896980463
0.9113693114122873	0.304786380165492	-0.26404953786921126	-0.9198093930452853	-0.26983848519899367	-0.03477462230664419	-0.03477462230664419	0.36092286665981144
-0.6774517497022479	0.528905447612025	0.9209664629352328	-0.634989016960501	0.48246891608780385	-0.29722100896980463	0.36092286665981144	-0.3794404541744525

Partición 2

-0.13067596295004535	0.5446653543182445	0.42307934586523344	-0.19710409601306234	0.41756704555887814	0.8331487776413274	0.468105182034913	-0.7804616370524962
0.5446653543182445	0.0949544066243011	-0.5383925252546216	-0.1155619949040545	0.6837378124987803	0.15440244258260194	0.6809251607989792	0.9680891550206352
0.42307934586523344	-0.5383925252546216	0.5859054089043414	0.2716205445292589	-0.039858827616618075	0.11753587599024751	-0.4436490490270506	0.5746605209245903
-0.19710409601306234	-0.1155619949040545	0.2716205445292589	0.5589755883533567	0.9358278004958743	0.11321137563874295	0.17901811537794177	0.49813136193511465
0.41756704555887814	0.6837378124987803	-0.039858827616618075	0.9358278004958743	0.6255115973691842	0.5906454227514378	-0.796106941656729	0.8908223677592784
0.8331487776413274	0.15440244258260194	0.11753587599024751	0.11321137563874295	0.5906454227514378	0.9812606865250112	0.057970239455491734	0.3944417065590293
0.468105182034913	0.6809251607989792	-0.4436490490270506	0.17901811537794177	-0.796106941656729	0.057970239455491734	0.9608366011321103	-0.45157757285159406
-0.7804616370524962	0.9680891550206352	0.5746605209245903	0.49813136193511465	0.8908223677592784	0.3944417065590293	-0.45157757285159406	0.8400822093406691

Partición 3

-0.26243981114655357	0.82227888026175	0.12775999606003163	0.8095958520974351	0.5613628927918307	0.8796388437459965	0.9918682736262406	0.9609216257614934
0.82227888026175	0.9451761524916653	0.7836461899322045	-0.8181506663612881	-3.5180933695933003E-4	-0.015279252137030075	-0.8727621870077562	0.6343241200682008
0.12775999606003163	0.7836461899322045	0.654897136291106	-0.9525220283644287	-0.819690769165254	0.26068481569889934	0.7872837551468885	0.6873077399158085
0.8095958520974351	-0.8181506663612881	-0.9525220283644287	0.6817512773926012	0.30457643938100354	0.14014447382120587	0.7809729248603094	-0.19859138524275277
0.5613628927918307	-3.5180933695933003E-4	-0.819690769165254	0.30457643938100354	0.047707061385359406	0.07219302588761645	0.6315664266575229	0.5899136218066815
0.8796388437459965	-0.015279252137030075	0.26068481569889934	0.14014447382120587	0.07219302588761645	0.8515181817584173	-0.14061592047826077	-0.06886502719556009
0.9918682736262406	-0.8727621870077562	0.7872837551468885	0.7809729248603094	0.6315664266575229	-0.14061592047826077	0.07777260202762715	0.5348409217514865
0.9609216257614934	0.6343241200682008	0.6873077399158085	-0.19859138524275277	0.5899136218066815	-0.06886502719556009	0.5348409217514865	0.5760347881778372

Partición 4

0.9143880033970591	0.5843864880473252	-0.16413148141132727	-0.40056757277602784	0.6122739565115084	0.9300096099528512	-0.5377744506893045	0.936801651734102
0.5843864880473252	-0.6497931484493729	0.35571449339274364	0.4976949002525092	0.12995740830940872	0.5036228006390104	0.7845688690927914	-0.8950567479807214
-0.16413148141132727	0.35571449339274364	-0.08635303943884887	-0.29521040536885246	0.5559743724531684	0.3067076707300823	0.3156364849933675	0.37768700143613554
-0.40056757277602784	0.4976949002525092	-0.29521040536885246	0.04130849711602158	0.2683584795479186	-0.4843622845796891	0.96382535479126	-0.23190525951743424
0.6122739565115084	0.12995740830940872	0.5559743724531684	0.2683584795479186	-0.5901017794022793	0.1067449423527758	0.799705389707541	0.3675803641330002
0.9300096099528512	0.5036228006390104	0.3067076707300823	-0.4843622845796891	0.1067449423527758	-0.8471407478431818	0.0739234710594967	-0.2888066104580371
-0.5377744506893045	0.7845688690927914	0.3156364849933675	0.96382535479126	0.799705389707541	0.0739234710594967	-0.16746414571395956	0.9393425109671723
0.936801651734102	-0.8950567479807214	0.37768700143613554	-0.23190525951743424	0.3675803641330002	-0.2888066104580371	0.9393425109671723	-0.12932493481044172

2.2. Matriz Diagonal

Partición 0

-0.5157080807846277	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.31995055738677336	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-0.9689703969538062	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.23948560499344662	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.012078372947042615	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.8492535822497402	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.15816049761467266	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.19140209909152905

Partición 1

0.6476189703321102	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-0.25442609989940057	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.20897395267584817	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.2366200827764618	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.26325525550109663	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.19297274876740445	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	-0.6304834308361003	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9650342516342951

Partición 2

0.536572176004173	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.08267083132590514	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-0.0037496760679853747	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-0.08089014898118285	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.01150978517431911	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.09701194334746044	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.8458783991552321	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.5100592603457287

Partición 3

0.3498436922710628	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-0.2038871646914917	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.797766809147641	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.07177874686961427	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.047209529891087176	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.9784568503149279	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.5508747155343866	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.1539867086221577

Partición 4

0.2559865929221955	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.587468912087904	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.7691216348600747	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-0.36479077604960775	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.03949616994996097	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.9650784395265481	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.5452142442476935	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.39882873003588215

2.3. Matriz Completa

Partición 0

0.6154178207454406	0.9237465195235525	0.2936834846299351	0.7306646669017088	0.4017628700423581	0.7992475771582621	0.6291339406061323	-0.3878235076292891
-0.7785814729219802	0.04316465021485594	0.5114171410139184	-0.4171285612092289	0.40489657721652605	0.982553523426325	0.5768573738781446	0.8770185832358033
-0.8044071542036579	-0.37135940474289675	0.24343771160802774	0.7097819372160357	-0.12163301872344634	0.3688171994649715	-0.5465004445565156	-0.49123264159844926
0.5523498397461153	0.29278750206589965	0.4921816285667082	0.7641702746257408	-0.717065661416651	0.3176665224699855	-0.6079467944885238	-0.6753076320164488
0.7513089584656226	0.5227999324326119	0.12432670329530993	0.33741350989585084	0.3055368412766287	0.3375016858778064	0.9968229070256348	-0.40932198391780306
-0.11805857123794783	0.9313339670970784	0.6772993653427726	-0.5984536678170203	0.4941568663114193	0.7244849596937216	0.414802677041355	0.9772266244818719
0.01638604825578338	0.9900587041274046	-0.24289169607820993	0.9610671075248312	0.3676500037457846	0.6147476217278858	0.24123029312665611	0.9199475376854518
0.1481125829177239	0.7762847934484924	0.2838808335960036	0.3533703206012314	0.9230739479113989	-0.2554430356341817	-0.14164357637345415	0.5251985657543344

Partición 1

0.9582888020773795	0.8411248436943304	0.5608895228066652	0.8365665262355962	0.17982646339647423	0.7557592999573405	0.18351894133774782	0.8678999163751713
-0.3795744775685066	-0.6645068827541221	0.41804186246640573	-0.7474666232757374	-0.8252652948086576	-0.4950852183384802	-0.7156612798440112	-0.8959496598633341
-0.22876760838393073	0.03423407546429047	-0.9708256686427259	-0.5016585906466393	-0.5458213738924789	-0.2867684280628795	0.4683860588836801	-0.35139760832656575
0.2830031253365899	-0.6824960637573523	0.2700233362492981	0.4095563395498174	0.07688829544456977	-0.20297152300756882	-0.6459969866110229	0.23346363061195063
-0.9431469647787392	0.3048709567967576	-0.7265975762423331	0.19128140847163078	-0.8892593094312593	0.2645841161285274	-0.29388652992919484	0.1061526362589229
-0.5837211001773912	-0.2997656692797297	-0.1484031315595411	-0.2219547841204781	-0.020353820940201217	0.3908859949020611	-0.994009531346848	0.18073198688942205
-0.2836791912732427	0.7008642531120604	-0.4495192068459877	0.5181026033577254	0.6378349713052577	-0.15076634395815436	0.8697526707271757	0.7103293102689472
-0.6410984236815143	-0.49314111955955797	0.6992933705360229	0.8297895053204678	-0.6874631750294593	0.1695842420545337	0.8886270000700576	0.9822668246461783

Partición 2

-0.8872316615297657	0.3714744107351162	0.5738072932892986	0.7258547532636717	0.8789153976512729	0.06635062010693993	0.8268800291898907	0.7712426470274185
-0.26169348615674526	-0.6417084499987848	0.8159523659372664	-0.5915216505582048	0.3371810609852137	0.7227905958358849	-0.43199376161555225	0.3707641425557282
0.9846840520414226	0.8598667199903967	0.07092100168690352	0.46642397859609064	-0.5072098013137467	-0.31212119831228036	0.7635880605549692	-0.795772878794458
0.5552504982908192	0.5032497439672572	0.2735526882397292	0.3603202611033629	0.40932643719826056	-0.9169525891812971	0.32707730330451734	0.8854662226698952
0.466278103845035	0.2212198721304225	0.5645336193074719	0.9478117501378409	0.7073182994496288	0.8963700410759811	0.4752512029118542	0.10289183334587859
-0.12938070700088533	0.7782641390771734	0.7004008680242206	-0.16089035142106556	0.7828054315736237	0.7742783946192487	-0.1265265056419247	-0.2914876502734116
-0.24030337184202533	0.25061895369914633	0.7414238681433071	0.1130385622407824	0.0014010577495167098	0.3782249340745045	0.4977400058199887	0.31823204583527176
0.34789269601246076	0.6553515913562045	0.005825649567276714	-0.8231440131734722	-0.7924143624165924	-0.17621097206363712	-0.029725460859748143	-0.12127043769381163

Partición 3

0.8815055773340323	0.15420911371965773	0.9004317062121088	-0.33935112073259543	-0.3224787078292026	-0.01927220147047759	-0.03652538533917793	0.9015958966880571
-0.361510677168128	0.8730090153209831	0.46780030597283107	0.2470232387925652	-0.024439834438713226	0.8974783502142283	-0.28233990542305365	-0.7814751722876357
0.7589799067327033	0.4708749922518344	0.8659210951120739	0.44540455416851676	0.0611622414853108	0.44252481019154366	-0.2754918147298562	0.79462844171716
-0.8526446413451088	0.23714546989856766	0.8178768830469261	0.15375489610525617	0.7081736428039213	0.8240518071253764	-0.551993100509172	-0.11213732277182664
0.4008191907226417	0.10334659453229944	-0.1376013247333816	-0.13540172210694013	-0.6266952350190438	-0.082564901065875	0.39135051201217059	-0.0011549483071010558
0.6526346396904159	0.5346383012248678	0.9494749978994855	0.8899681753609324	-0.7604733900589234	-0.15457283559946156	0.6063141718040422	0.5796026398469334
0.9165062952582956	0.11994301792835704	0.314413048224671	-0.21364060922100447	-0.7766750078180396	0.19425167431558688	-0.11018589634231735	0.3855737986855239
-0.2137045633071053	-0.7461739865964239	0.216093733253363	-0.020405740533358974	0.44241440893795936	0.6941821580728428	0.5812480651016438	0.9210820674686675

Partición 4

0.8314550712075814	-0.29137503012003807	0.5949714885594583	0.7670617704407988	0.9773940154081188	0.9151328476238739	-0.8780527006388067	-0.8743388441141162
0.8677584967880487	-0.9002239138070827	0.7425301808817887	-0.9445550976550904	-0.1731455599770344	0.9770456219390293	0.8438494246423351	-0.245987065849959
-0.4467113584863509	0.5892340610283021	0.8201857291714166	0.3149635852475219	0.8948244899179729	-0.10356388920856929	-0.42691218445017465	-0.15537085304894127
-0.039100149723521604	-0.28936978097362465	0.8110518704115806	-0.6475559075229242	0.8694206831370509	-0.2876290677689206	0.038922904071048925	-0.4079425514604038
-0.23225674066073454	0.9895974329488298	0.8723033484635283	0.6037476643003696	0.11802349312295835	0.607800557207336	0.8574128414752432	-0.4638418683985064
0.30218635352484646	-0.799512406176546	-0.057792595750446485	0.4327380725509471	0.7173174712552979	-0.7687167469502479	0.2742746331882945	0.06496653555033716
0.1854249445404299	0.3524285418399644	-0.579988600002504	0.7366296635482609	-0.3320016851421922	0.06531397773438286	-0.5374121412889667	-0.41295360063794084
0.18100710423315713	-0.0985756360308856	0.1742619329708186	0.4199829866757794	0.6215983229197531	0.5541027540900101	0.1090988931623376	0.5016495391197614

3. Matrices resultado del dominio Rectas Ruido 0 grados

Recordamos los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	9,44%	65,00%	100,00%	90,00%	100,00%	95,00%	100,00%	85,00%
1	8,33%	55,00%	99,44%	80,00%	100,00%	100,00%	100,00%	80,00%
2	5,56%	30,00%	100,00%	70,00%	100,00%	70,00%	100,00%	65,00%
3	10,00%	15,00%	100,00%	65,00%	100,00%	80,00%	100,00%	80,00%
4	8,89%	50,00%	100,00%	85,00%	100,00%	90,00%	100,00%	75,00%
5	8,89%	35,00%	100,00%	80,00%	100,00%	95,00%	100,00%	80,00%
6	9,44%	35,00%	100,00%	60,00%	100,00%	60,00%	100,00%	60,00%
7	10,00%	45,00%	100,00%	80,00%	100,00%	95,00%	99,44%	80,00%
8	7,22%	65,00%	100,00%	85,00%	100,00%	95,00%	100,00%	85,00%
9	6,11%	50,00%	100,00%	80,00%	100,00%	80,00%	100,00%	100,00%
MEDIA	8,39%	44,50%	99,94%	77,50%	100,00%	86,00%	99,94%	79,00%
VARIANZA	0,03%	2,52%	0,00%	0,90%	0,00%	1,71%	0,00%	1,21%

Tabla 24: Resultados del dominio Rectas Ruido 0 grados

3.1. Matriz Simétrica

Partición 0

$$\begin{bmatrix} 0.022360726547881127 & -3.698307510804355E-4 \\ -3.698307510804355E-4 & -0.34401436040901145 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.03430735663552387 & 0.030557279586512553 \\ 0.030557279586512553 & -0.6895749253710621 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.02795373342913643 & -0.0017442048901583629 \\ -0.0017442048901583629 & 0.42260972561608057 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.050886105263825865 & -0.01699763041308977 \\ -0.01699763041308977 & -0.7934364472171076 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.0543035746100724 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} 0.04352820125265677 & -0.04014242571659987 \\ -0.04014242571659987 & 0.92649780215368 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} 0.05019240930913349 & -0.03360714765298578 \\ -0.03360714765298578 & 0.5639370423716558 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} -0.017622246411906906 & 0.01718766470960209 \\ 0.01718766470960209 & -0.9614050319029868 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} -0.04028190616516092 & -0.02253597509190164 \\ -0.02253597509190164 & 0.7272736295202966 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} -0.08056991709227157 & -0.0023476240311524066 \\ -0.0023476240311524066 & 0.9587355567905976 \end{bmatrix}$$

3.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} -0.031694201084505824 & 0.0 \\ 0.0 & 0.7496094879364821 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.001012094678964126 & 0.0 \\ 0.0 & 0.4795551379553834 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.05372819722326194 & 0.0 \\ 0.0 & 0.8944959288775084 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.022833028940250966 & 0.0 \\ 0.0 & 0.5006142841008635 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.040149465698736786 & 0.0 \\ 0.0 & 0.9954045679548791 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} 0.02469371736526116 & 0.0 \\ 0.0 & -0.7463680721188964 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} -0.07765291392194751 & 0.0 \\ 0.0 & 0.8990175130329416 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} -0.02180553858241152 & 0.0 \\ 0.0 & 0.9880394627584734 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} 0.027879218394000427 & 0.0 \\ 0.0 & 0.6969064868546528 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} 0.05734899689950179 & 0.0 \\ 0.0 & -0.7266875024899668 \end{bmatrix}$$

3.3. Matriz Completa

Partición 0

$$\begin{bmatrix} -0.047540503922165134 & 0.02486662182574373 \\ 0.7670183908961121 & 0.39338464764849346 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.018006328993079634 & 0.0194385612488861 \\ 0.8067628584941711 & -0.35049934980638264 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.07887985818704801 & 0.0685894733499619 \\ 0.756160881704026 & 0.9751914074114006 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} -0.05021231789980607 & 0.033013941663483815 \\ -0.6308105166654472 & -0.846196396310205 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.031257126404773294 & -0.03268305936128013 \\ 0.5059106122385957 & -0.21390877788183116 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} -0.05685705966837573 & -0.018001781084004076 \\ 0.4042639331633664 & 0.966793327870695 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} 0.03009449368925532 & 0.11911452370041964 \\ 0.8864631361679675 & -0.8843083523862034 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} 0.04875057771976221 & -0.047644142662123246 \\ 0.6076674241388736 & 0.825747518977711 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} -0.042071243071930775 & 0.03500249113178233 \\ -0.23805128806175502 & -0.8555706759866033 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} 0.021826414817684325 & 0.005558234556093611 \\ 0.20593889420809786 & -0.8365895882733596 \end{bmatrix}$$

4. Matrices resultado del dominio Rectas Ruido 45 grados

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	8,89%	35,00%	100,00%	90,00%	10,00%	35,00%	99,44%	95,00%
1	7,78%	35,00%	100,00%	90,00%	9,44%	35,00%	100,00%	35,00%
2	10,00%	50,00%	100,00%	100,00%	10,56%	50,00%	100,00%	100,00%
3	7,78%	45,00%	100,00%	95,00%	8,33%	45,00%	99,44%	55,00%
4	7,22%	55,00%	100,00%	90,00%	10,00%	55,00%	100,00%	95,00%
5	7,78%	55,00%	100,00%	100,00%	8,89%	55,00%	100,00%	100,00%
6	8,33%	60,00%	99,44%	70,00%	13,33%	60,00%	99,44%	70,00%
7	9,44%	50,00%	100,00%	75,00%	10,00%	50,00%	100,00%	75,00%
8	9,44%	45,00%	100,00%	100,00%	10,56%	45,00%	99,44%	85,00%
9	10,00%	70,00%	100,00%	100,00%	11,11%	70,00%	100,00%	100,00%
MEDIA	8,67%	50,00%	99,94%	91,00%	10,22%	50,00%	99,78%	81,00%
VARIANZA	0,01%	1,17%	0,00%	1,16%	0,02%	1,17%	0,00%	4,93%

Tabla 25: Resultados del dominio Rectas Ruido 45 grados

4.1. Matriz Simétrica

Partición 0

$$\begin{bmatrix} 0.5169204321547946 & -0.5133621441198395 \\ -0.5133621441198395 & 0.4946192811224246 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.7584612583857406 & 0.8092593140468556 \\ 0.8092593140468556 & -0.8141450278838407 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.7393529282696472 & -0.7287826895478953 \\ -0.7287826895478953 & 0.7259898555031414 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.15767860622452334 & -0.15834360826848548 \\ -0.15834360826848548 & 0.152620222324529 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.7614618802167419 & 0.7559293473674955 \\ 0.7559293473674955 & -0.7316659995802564 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} 0.5467454580580625 & -0.5451315622666301 \\ -0.5451315622666301 & 0.5271942647106362 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} -0.5590154605094375 & 0.7781219178904211 \\ 0.7781219178904211 & -0.8258383425163811 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} -0.5461540696288759 & 0.7832110096869227 \\ 0.7832110096869227 & -0.8361581578822903 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} 0.6405143484332125 & -0.6200464162591257 \\ -0.6200464162591257 & 0.6249578970719103 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} 0.4879509123981116 & -0.48039506790692954 \\ -0.48039506790692954 & 0.4830358578138234 \end{bmatrix}$$

4.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} 0.5105577708275693 & 0.0 \\ 0.0 & 0.32079760295526927 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.9366454390970826 & 0.0 \\ 0.0 & -0.9154560642045557 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.7880401153269212 & 0.0 \\ 0.0 & 0.7644664869722013 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.7489211058104631 & 0.0 \\ 0.0 & 0.6999765368379757 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.3846597172762527 & 0.0 \\ 0.0 & 0.2170634796368509 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} 0.6006281122833512 & 0.0 \\ 0.0 & 0.40419255106306307 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} 0.7044111647874776 & 0.0 \\ 0.0 & 0.9625210265468169 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} -0.66358948540439 & 0.0 \\ 0.0 & 0.6435424002205721 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} 0.8945806770729905 & 0.0 \\ 0.0 & 0.8639145697411005 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} -0.5047793018457469 & 0.0 \\ 0.0 & 0.6899970441304015 \end{bmatrix}$$

4.3. Matriz Completa

Partición 0

$$\begin{bmatrix} -0.5852850365337028 & 0.8361639758730341 \\ 0.6246564064410463 & -0.8176475009938795 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.1217026194994345 & 0.6964952802290983 \\ -0.29516875189647573 & -0.7694619012731456 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.5088242542789378 & -0.8715656646976067 \\ -0.48324883430398047 & 0.8688248808513022 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.23110115039555934 & -0.583502182008882 \\ -0.164256877680252 & 0.7215365865548358 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.745037890523371 & -0.14085400880827614 \\ 0.7271173365127022 & 0.08743298287986678 \end{bmatrix}$$

Partición 5

$$\begin{bmatrix} 0.6193485494600467 & -0.9816054530987155 \\ -0.6133644388872046 & 0.9848414730958093 \end{bmatrix}$$

Partición 6

$$\begin{bmatrix} -0.18820716125489467 & -0.7640864814175282 \\ 0.3181631759456309 & 0.8442239798399336 \end{bmatrix}$$

Partición 7

$$\begin{bmatrix} -0.7033119820130146 & 0.2996038505002342 \\ 0.9098524830557362 & -0.24950227064584307 \end{bmatrix}$$

Partición 8

$$\begin{bmatrix} 0.6587931077872203 & -0.07047965407843337 \\ -0.6790627266370586 & 0.03794067841256832 \end{bmatrix}$$

Partición 9

$$\begin{bmatrix} 0.7978937939021249 & 0.9194530674006549 \\ -0.7979897542295316 & -0.9015777725104277 \end{bmatrix}$$

5. Matrices resultado del dominio Rectas4D-45g

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	0,00%	0,00%	100,00%	97,50%	5,63%	7,50%	100,00%	97,50%
1	0,63%	0,00%	100,00%	100,00%	6,25%	5,00%	100,00%	97,50%
2	0,63%	2,50%	100,00%	90,00%	6,88%	2,50%	100,00%	92,50%
3	1,25%	5,00%	100,00%	95,00%	6,88%	7,50%	99,38%	95,00%
4	0,63%	5,00%	100,00%	97,50%	6,25%	5,00%	100,00%	100,00%
MEDIA	0,63%	2,50%	100,00%	96,00%	6,38%	5,50%	99,88%	96,50%
VARIANZA	0,00%	0,06%	0,00%	0,14%	0,00%	0,04%	0,00%	0,08%

Tabla 26: Resultados del dominio Rectas4D 45 grados

5.1. Matriz Simétrica

Partición 0

$$\begin{bmatrix} 0.13327335771304338 & -0.033372280115544495 & 0.38825237166333426 & -0.4098427918818296 \\ -0.033372280115544495 & -0.8357224214419877 & 0.21968077995550017 & 0.972608353796323 \\ 0.38825237166333426 & 0.21968077995550017 & 0.6764559692058087 & -0.8990058827547325 \\ -0.4098427918818296 & 0.972608353796323 & -0.8990058827547325 & 0.23999132216175545 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.7758858741088261 & -0.4573975639562351 & -0.15535912316176392 & -0.0983162457154958 \\ -0.4573975639562351 & 0.23086015255817027 & 0.7130256146680893 & -0.5830805703206527 \\ -0.15535912316176392 & 0.7130256146680893 & -0.4725484775820046 & -0.12337663117454023 \\ -0.0983162457154958 & -0.5830805703206527 & -0.12337663117454023 & 0.7752114432268242 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.11281447294126923 & 0.38204463107323927 & 0.3711226304531767 & -0.6575915169980961 \\ 0.38204463107323927 & 0.4718315520353187 & -0.917760048333446 & 0.1326489219410354 \\ 0.3711226304531767 & -0.917760048333446 & -0.3797133210269923 & 0.8691114137052456 \\ -0.6575915169980961 & 0.1326489219410354 & 0.8691114137052456 & -0.17827449966024078 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.6217505826531922 & -0.15419047007481024 & 0.49586028248964553 & -0.9269252477454635 \\ -0.15419047007481024 & 0.7292019130771371 & -0.354452642302878 & -0.47781063542170665 \\ 0.49586028248964553 & -0.354452642302878 & -0.8805290189346164 & 0.8688984861303009 \\ -0.9269252477454635 & -0.47781063542170665 & 0.8688984861303009 & 0.31028961916573083 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.22255834318171475 & -0.080185849917352 & 0.11826659100583503 & 0.47556395995048173 \\ -0.080185849917352 & 0.9452527738546355 & 0.12796968311841805 & -0.7019294339946831 \\ 0.11826659100583503 & 0.12796968311841805 & 0.39456421869574365 & -0.5512008975108411 \\ 0.47556395995048173 & -0.7019294339946831 & -0.5512008975108411 & 0.7392851949539561 \end{bmatrix}$$

5.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} -0.029280461478421227 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2018779142192706 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.8438519224958383 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.9082954784773671 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.06727196633328492 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.05230665429045811 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.6475411096173873 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.762562943486897 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.03075964945456411 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.12111648187853663 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.6584801362525936 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.6998662457373275 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.06046925034014228 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.04880412217499064 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.36199812235126916 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.47634967892030344 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.16395718682289384 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.7345742248740135 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.05189189351591572 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7952026255909906 \end{bmatrix}$$

5.3. Matriz Completa

Partición 0

$$\begin{bmatrix} -0.04639383089114435 & -0.2555609762580169 & 0.25195477451291826 & 0.011216587045334125 \\ -0.3050243533901643 & 0.4257153728369153 & 0.3288365351888801 & 0.7838397852533654 \\ 0.05500967356685038 & -0.43978803974879 & 0.3474220505586363 & -0.09459433967891095 \\ 0.24331148786995827 & 0.5183241571453304 & -0.5213734118195961 & -0.6810763277216529 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.2599063055212345 & 0.23882203760497744 & -0.9367227871348476 & -0.9328979152311363 \\ 0.4352161030619075 & 0.7352749972701682 & 0.4606907739014483 & -0.09304791406840063 \\ 0.43244157725503674 & -0.43234059614119547 & -0.24950869888834215 & 0.4928756821585636 \\ -0.9162195836588118 & -0.8100108312954235 & 0.9914116731234424 & 0.7081871649538269 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.05062740155609302 & 0.025471248279599568 & -0.37208320061630995 & -0.30265529650912226 \\ 0.1572786511554245 & 0.3921244633210641 & -0.13062817388267645 & -0.2843562765613231 \\ -0.6237338076543371 & 0.13904864722634147 & 0.5100019640552407 & -0.4588703567370638 \\ 0.5042995784626518 & -0.7602705314884126 & -0.4646260172014183 & 0.9779760060907343 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} -0.6491878017510644 & -0.4893273384418988 & 0.43671179998561316 & -0.7898281160529161 \\ -0.1577093077384466 & 0.8919376079790169 & 0.10358290749326826 & 0.6997931276865104 \\ 0.31938773041361146 & 0.4265077261487542 & 0.2022819616586401 & -0.35088423089945975 \\ 0.6092605730892637 & -0.6403055883245135 & -0.4474454194746811 & 0.6098778741224156 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.15160543085552325 & 0.6193556543913741 & -0.2037991148157694 & -0.6906460938927561 \\ -0.06121385769657972 & 0.5812274625047396 & -0.3258019128117824 & 0.8261544253531004 \\ 0.5560915192975391 & -0.3973074496622657 & -0.5180765824294908 & 0.2044692152404124 \\ 0.11183349766028683 & -0.888920980025915 & 0.9138825442680605 & -0.20560564157449512 \end{bmatrix}$$

6. Matrices resultado del dominio Aleatorio

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	91,67%	83,33%	99,17%	93,33%	97,50%	91,67%	99,17%	88,33%
1	91,67%	90,00%	99,17%	96,67%	97,50%	93,33%	99,17%	91,67%
2	92,08%	76,67%	99,17%	93,33%	97,50%	95,00%	99,58%	91,67%
3	90,83%	86,67%	98,33%	91,67%	97,92%	100,00%	99,58%	90,00%
4	89,17%	81,67%	97,92%	90,00%	98,33%	93,33%	98,75%	95,00%
MEDIA	91,08%	83,67%	98,75%	93,00%	97,75%	94,67%	99,25%	91,33%
VARIANZA	0,01%	0,26%	0,00%	0,06%	0,00%	0,10%	0,00%	0,06%

Tabla 27: Resultados del dominio Aleatorio

6.1. Matriz Simétrica

Partición 0

0.8501617320143009	-0.9622664920592543	-0.4453791212925463	-0.7749840747351274
-0.9622664920592543	0.7890129558672966	0.5355295429957704	0.7535996507090034
-0.4453791212925463	0.5355295429957704	0.08820594429861361	-0.06050663230707709
-0.7749840747351274	0.7535996507090034	-0.06050663230707709	-0.01581886936141519

Partición 1

0.7947599349072374	-0.583654731992181	-0.5489002650666097	0.18068830985585405
-0.583654731992181	0.6307393435793274	-0.17329859125238345	0.0
-0.5489002650666097	-0.17329859125238345	0.5128233379234953	-0.16279049720088
0.18068830985585405	0.0	-0.16279049720088	0.42987605649379435

Partición 2

-0.8398166338451372	0.9127957196973702	-0.5190620578641154	-0.3435191887221971
0.9127957196973702	-0.9264033352813434	0.5308627762049257	0.5509981883260233
-0.5190620578641154	0.5308627762049257	0.5223787561808063	0.07139904441112188
-0.3435191887221971	0.5509981883260233	0.07139904441112188	-0.8892347596052381

Partición 3

0.9727947393836844	0.9957491636245388	-0.7118676679399081	-0.7159991966984844
0.9957491636245388	-0.9035411976446213	0.5171961825401006	0.22242741836048474
-0.7118676679399081	0.5171961825401006	0.840521434512357	0.5401377456075847
-0.7159991966984844	0.22242741836048474	0.5401377456075847	0.29050083586704667

Partición 4

$$\begin{bmatrix} -0.7345534090137136 & 0.6480279130015306 & 0.6246453897831797 & 0.07926288987835473 \\ 0.6480279130015306 & -0.4197418707132963 & -0.753735538778401 & 0.24301537435332432 \\ 0.6246453897831797 & -0.753735538778401 & 0.46894308504702364 & -0.08117307016419395 \\ 0.07926288987835473 & 0.24301537435332432 & -0.08117307016419395 & -0.6657941492697257 \end{bmatrix}$$

6.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} 0.4515232917086456 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.9736565833222046 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.008081252208309575 & 0.0 \\ 0.0 & 0.0 & 0.0 & 2.88133607626917E-4 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.3778960079833953 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.6901060920518303 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.05408914544393073 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.06370282023521112 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.32042451476368405 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5680739481733361 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.0017005280485222762 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.01121033729999421 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.36777321626064596 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.8480377100534435 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.01725950170455759 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.004323447012443249 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.963848644216314 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.7935768064331823 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.022335445524600983 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.03019516745466788 \end{bmatrix}$$

6.3. Matriz Completa

Partición 0

$$\begin{bmatrix} 0.8147090111184034 & -0.2231513972893414 & -0.7011180423874304 & -0.03424340890615385 \\ -0.13449704462167666 & 0.5101232152028421 & 0.9737594953502182 & -0.1741925525236545 \\ 0.3413739560757658 & 0.20281717034287894 & 0.12323248353702743 & 0.5960286108339485 \\ -0.19376860232665694 & -0.14403552032410838 & -0.03003935691939441 & -0.29700737545419975 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} 0.6469063387571956 & 0.35038574859832794 & -0.9330640765723246 & -0.6511319616014124 \\ -0.8204516267054605 & -0.2711077634037977 & 0.8180516164106967 & 0.6802825521724932 \\ -0.011259212500656757 & 0.683619888117049 & -0.48212164899445575 & -0.1033257568814856 \\ 0.2457738629950561 & -0.32623261933257464 & 0.3531600837047789 & 0.14272645128363282 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.27835234455216806 & 0.6827383053895311 & -0.708755687935513 & -0.6832867651940163 \\ -0.3350056727732542 & -0.7098934371423935 & 0.5351456993836755 & 0.7890767630238785 \\ 0.7429616704423752 & 0.12127167424777441 & 0.6415171614845524 & 0.4901947772908316 \\ -0.15470055224527313 & 0.6723567577980283 & 0.22498945241760904 & 0.5033577988937472 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} -0.8519730103365264 & 0.14384583829240516 & -0.36485790439206944 & 0.7171754007379426 \\ 0.8819888758958084 & -0.07589960679049856 & 0.2563827479991636 & -0.48595494132791517 \\ 0.22018021792650355 & 0.0032692298065117864 & 0.03592784111978742 & 0.9389560503004764 \\ -0.03807833928766591 & 0.0016755709914539896 & 0.07096283438194151 & 0.071117411521336 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.7254342081113494 & 0.868100522310933 & -0.9808514889567816 & -0.3804445388256543 \\ 0.9207199130948129 & -0.35310933814855683 & 0.5741689688624335 & 0.0061729566631296695 \\ -0.2982199453207105 & -0.52215367632501 & 0.7690213619818961 & 0.3698608432546415 \\ -0.2876523292146673 & -0.3673104855162639 & 0.7122740221983961 & 0.5808066144120951 \end{bmatrix}$$

7. Matrices resultado del dominio Aleatorio Girado

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	88,33%	85,00%	95,42%	81,67%	93,33%	83,33%	95,83%	81,67%
1	86,25%	86,67%	95,00%	86,67%	90,42%	86,67%	96,67%	91,67%
2	89,17%	78,33%	95,83%	78,33%	91,67%	81,67%	95,00%	78,33%
3	89,17%	83,33%	97,08%	86,67%	92,50%	83,33%	96,25%	88,33%
4	83,33%	86,67%	95,42%	86,67%	88,75%	80,00%	95,00%	93,33%
MEDIA	87,25%	84,00%	95,75%	84,00%	91,33%	83,00%	95,75%	86,67%
VARIANZA	0,06%	0,12%	0,01%	0,15%	0,03%	0,06%	0,01%	0,42%

Tabla 28: Resultados del dominio Aleatorio Girado

7.1. Matriz Simétrica

Partición 0

$$\begin{bmatrix} -0.46875829269780667 & 0.3277165839036468 & -0.16365456320012228 & 0.6232990711562054 \\ 0.3277165839036468 & 0.5152946587163069 & 0.07850683235861455 & -0.18939653575879234 \\ -0.16365456320012228 & 0.07850683235861455 & 0.6152444226081089 & 0.034209855205725725 \\ 0.6232990711562054 & -0.18939653575879234 & 0.034209855205725725 & -0.2589514634535607 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.4921651945998212 & 0.6760956858277755 & -0.9286229630356369 & 0.7883799636106723 \\ 0.6760956858277755 & 0.39486898111614277 & 0.2059054749152105 & -0.6760091060813438 \\ -0.9286229630356369 & 0.2059054749152105 & 0.9975402206220502 & 0.28944725000702853 \\ 0.7883799636106723 & -0.6760091060813438 & 0.28944725000702853 & -0.6250335017613218 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.5912188694900053 & 0.020246807939616085 & 0.8321437171098399 & -0.2902528830627651 \\ 0.020246807939616085 & 0.3570010406166013 & -0.03483553708964482 & 0.7524927247233775 \\ 0.8321437171098399 & -0.03483553708964482 & -0.2806566795193828 & -0.7096685008312997 \\ -0.2902528830627651 & 0.7524927247233775 & -0.7096685008312997 & 0.3310861193436645 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.23378558688176776 & -0.223871322949049 & 0.8000896172580457 & -0.7639013260219925 \\ -0.223871322949049 & 0.18652773094364206 & 0.2515970029164023 & 0.2814115027967693 \\ 0.8000896172580457 & 0.2515970029164023 & 0.16714503985862472 & -0.31222689156887595 \\ -0.7639013260219925 & 0.2814115027967693 & -0.31222689156887595 & 0.8850070826371519 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} -0.9516599167357893 & -0.1644146277612748 & -0.3994834383229695 & 0.7419851042125215 \\ -0.1644146277612748 & -0.7698932968420826 & 0.4673693631554503 & -0.020514725877534445 \\ -0.3994834383229695 & 0.4673693631554503 & 0.460160287878397 & 0.3050358198784364 \\ 0.7419851042125215 & -0.020514725877534445 & 0.3050358198784364 & -0.7000493153131992 \end{bmatrix}$$

7.2. Matriz Diagonal

Partición 0

$$\begin{bmatrix} -0.7468186319307555 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.888579572334645 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.6254548213516156 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.09645813348247878 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.9947918720035855 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.6771409283807075 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.9270267496366762 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.08102460367936382 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} 0.8670625059007471 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.28861463822112154 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.30841351938276695 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.4779463881685292 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.5131787048091667 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3879427055009683 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.9416689055825691 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.008508649128400325 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.7582444923800595 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.9944345433503416 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.4634159944750287 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.14125454789241054 \end{bmatrix}$$

7.3. Matriz Completa

Partición 0

$$\begin{bmatrix} 0.9130543698505638 & 0.7807715377385155 & -0.5641048767067919 & -0.4517937625255648 \\ -0.13258654371317036 & 0.625113310019184 & 0.5631188630076387 & 0.746002565223796 \\ -0.03898267293401314 & 0.25017108207839855 & -0.4205360381232732 & -0.05357763552456274 \\ -0.6746225652964889 & -0.42579835728991533 & 0.5451115468987173 & 0.740426460928269 \end{bmatrix}$$

Partición 1

$$\begin{bmatrix} -0.5451666635025771 & -0.837535113084025 & -0.10214557504122801 & -0.6711536051929192 \\ 0.39772674503072114 & 0.1403599772785995 & 0.049569724638552914 & 0.0898939865966264 \\ 0.08441613805087789 & -0.4830431664021819 & 0.48515227446100184 & 0.13448878798537178 \\ 0.41489324155932406 & 0.9168391669028838 & -0.47060146576178186 & 0.6351196279380219 \end{bmatrix}$$

Partición 2

$$\begin{bmatrix} -0.8970327379931042 & -0.8860202859630459 & 0.9141580374807542 & 0.3143475913081434 \\ 0.7548377783130988 & 0.7382035017920429 & 0.20202593107100775 & 0.6775184012699585 \\ -0.8174980791472333 & -0.2784187799213038 & 0.2689951451170669 & -0.9262588125664799 \\ 0.6271205390186247 & 0.9913532801866863 & -0.244901096728819 & -0.03576105653647377 \end{bmatrix}$$

Partición 3

$$\begin{bmatrix} 0.6518336625002633 & -0.739175838477228 & 0.38570237689093667 & -0.18718851975445372 \\ 0.2057704081199797 & 0.25831010068735416 & -0.30361340323897734 & -0.016117793010744896 \\ 0.3223230337370826 & -0.5999773962352918 & -0.48662083952089064 & 0.5951967938075352 \\ -0.515448939736045 & 0.9381465944378838 & -0.34516415825657953 & -0.9313525927446115 \end{bmatrix}$$

Partición 4

$$\begin{bmatrix} 0.8722401736006241 & 0.2889928992987506 & -0.43112265547236184 & -0.8502768239807623 \\ 0.20467739807305074 & -0.16993291924482012 & 0.25385804516462207 & 0.8699940471813478 \\ -0.4828573167728809 & -0.005970353769047643 & 0.10656376587378236 & -0.6056329542589702 \\ -0.3335074930240225 & 0.12540641522450224 & 0.09109339859466603 & 0.7937173874657446 \end{bmatrix}$$

8. Matrices resultado del dominio EEG

8.1. Conjunto de datos con clases 2 y 3

Recordamos los resultados obtenidos en las experimentaciones:

Partición	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	83,83 %	90,02%	84,21%	88,71%	85,26%	90,07%	83,94%	90,02%

Tabla 29: Resultados del dominio EEG con clases 2 y 3

Matriz Simétrica

-0.24381301813995726	-0.7793663620323437	-0.6275380698111293	0.3731734536516569	0.524851344782428	0.05249627089691333
-0.7793663620323437	0.43499713034080045	0.23001301097181837	-0.4342757932093373	0.8748578602340361	-0.7912196028681027
-0.6275380698111293	0.23001301097181837	0.8355371168671324	0.3115085799443553	-0.09433912643163378	0.5693968235353952
0.3731734536516569	-0.4342757932093373	0.3115085799443553	-0.6061390497798536	0.6896681722152957	0.702000598114736
0.524851344782428	0.8748578602340361	-0.09433912643163378	0.6896681722152957	-0.35007651148060903	0.6399619875342452
0.05249627089691333	-0.7912196028681027	0.5693968235353952	0.702000598114736	0.6399619875342452	-0.9335575644746668

Matriz Diagonal

0.7397149691515329	0.0	0.0	0.0	0.0	0.0
0.0	0.6809170694279569	0.0	0.0	0.0	0.0
0.0	0.0	-0.26073894511382645	0.0	0.0	0.0
0.0	0.0	0.0	0.2028497172755367	0.0	0.0
0.0	0.0	0.0	0.0	-0.5017392657693374	0.0
0.0	0.0	0.0	0.0	0.0	0.9306870507032294

Matriz Completa

1.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0
0.6800590097215273	0.1731634757247681	0.8607024673619337	0.779721035519964	0.15143641550991926	0.9752210556448309

8.2. Conjunto de datos con clases 2 y 7

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	87,33 %	93,27%	88,32%	93,31%	88,91%	93,23%	87,33%	93,27%

Tabla 30: Resultados del dominio EEG con clases 2 y 7

Matriz Simétrica

$$\begin{bmatrix}
 -0.6382301163641351 & 0.4394388154987966 & 0.11810564066509777 & -0.4636571188916718 & -0.6856422265907907 & -0.9878184966724066 \\
 0.4394388154987966 & 0.7184258008567943 & 0.7332761352843299 & 0.0700233629910817 & 0.6547725088212749 & 0.4354538683943482 \\
 0.11810564066509777 & 0.7332761352843299 & -0.24972049183923062 & 0.037462200207735497 & 0.014988416422319628 & 0.1534908523104026 \\
 -0.4636571188916718 & 0.0700233629910817 & 0.037462200207735497 & -0.523758248275207 & 0.4685228502768627 & 0.46758758618849194 \\
 -0.6856422265907907 & 0.6547725088212749 & 0.014988416422319628 & 0.4685228502768627 & -0.5901385691589205 & 0.817346924540295 \\
 -0.9878184966724066 & 0.4354538683943482 & 0.1534908523104026 & 0.46758758618849194 & 0.817346924540295 & -0.3166523901241063
 \end{bmatrix}$$

Matriz Diagonal

$$\begin{bmatrix}
 -0.6698236638954379 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & -0.707315222351945 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.25752813625331705 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.5013291490384508 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.507545217215186 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7966756762026814
 \end{bmatrix}$$

Matriz Completa

$$\begin{bmatrix}
 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0
 \end{bmatrix}$$

No se mejoró el resultado de la matriz euclídea.

8.3. Conjunto de datos con clases 3 y 7

Recordamos los resultados obtenidos en las experimentaciones:

	Distancia Euclídea		Evolución Matriz Simétrica		Evolución Matriz Diagonal		Evolución Matriz Completa	
Partición	Prueba	Test	Prueba	Test	Prueba	Test	Prueba	Test
0	82,03 %	85,63%	82,99%	88,31%	84,53%	90,22%	82,03%	85,63%

Tabla 31: Resultados del dominio EEG con clases 3 y 7

Matriz Simétrica

1.0	0.8347953301112424	0.0368697098231181	-0.035160514751452876	-0.6425053404524375	0.270229830865087
0.8347953301112424	0.5000240104298611	0.0	-0.14942989871651524	0.6930528573873602	-0.8415313271736264
0.0368697098231181	0.0	-0.1384637525008119	0.48190026835724453	-0.32444153048435176	-0.4478662738425965
-0.035160514751452876	-0.14942989871651524	0.48190026835724453	0.27804731266457305	0.5262296727206637	0.41640116437073715
-0.6425053404524375	0.6930528573873602	-0.32444153048435176	0.5262296727206637	0.2542239707538857	0.6705956474056913
0.270229830865087	-0.8415313271736264	-0.4478662738425965	0.41640116437073715	0.6705956474056913	0.5054635968739969

Matriz Diagonal

0.5073359632416912	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.8558165273559645	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-0.19297582338096841	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-0.12000216819266385	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-0.7741662113359298	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.9131064992057318

Matriz Completa

1.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0

No se mejoró el resultado de la matriz euclídea.

ANEXO B: PRESUPUESTO

Este anexo contiene el presupuesto asociado al desarrollo del proyecto. Se detalla la estimación de los gastos que supondrá realizar este proyecto en un entorno universitario, estimando los costes y la duración del mismo.

Se detallaran los costes de personal y los costes de material, así como un resumen total del presupuesto.

ANEXO B: Presupuesto

1. Costes de Personal

El personal que ha realizado el proyecto es el siguiente:

CATEGORIA	Nº PERSONAS	COSTE/HORA
Programador	1	25 Euros
Analista	1	100 Euros
Expertos en IA	2	120 Euros

Tabla 32: Personal

Los costes de personal no incluirán beneficios, ya que se trata de un proyecto fin de carrera de una Universidad.

La duración del proyecto se estima que sea de 900 horas de trabajo, repartidas entre 200 horas para análisis, 300 horas de programación y 400 de experimentación y pruebas. Las tareas de experimentación y pruebas, las realizará el programador. Además, se requerirán de unas 100 horas de asesoramiento por parte de expertos en Inteligencia Artificial.

Por tanto, los costes asociados a personal ascenderían a:

CATEGORIA	Nº HORAS	COSTE
Programador	700	17.500 Euros
Analista	200	2.000 Euros
Expertos en IA	100	12.000 Euros
TOTAL	1000	31.500 Euros

Tabla 33: Costes de Personal

2. Costes de Material

El proyecto se desarrollará sobre un ordenador personal de sobremesa con las siguientes características:

- Intel Core 2 Quad Q9400 2.66GHz
- Sistema Operativo Ubuntu 10.04

Esta máquina está valorada aproximadamente en 650 Euros.

El Sistema Operativo que se utilizará será Ubuntu [UBUNTU, <http://www.ubuntu.com/>], cuyo coste es gratuito.

El IDE de desarrollo a utilizar será Eclipse [**ECLIPSE**, <http://www.eclipse.org/>], cuyo coste es gratuito. Además, el framework JGAP [**JGAP**, <http://jgap.sourceforge.net/>] que se utilizará para la codificación del algoritmo genético, también es gratuito.

Se han añadido también gastos de fungibles como papel, etc. El total de costes de material ascienden a:

MATERIAL	COSTE
PC	650 Euros
FUNGIBLE	50 Euros
TOTAL	700 Euros

Tabla 34: Costes de Material

3. Presupuesto Final

A continuación se calcula el coste total del proyecto, teniendo en cuenta el coste de los recursos humanos y materiales necesarios para la realización del mismo, así como los riesgos y gastos indirectos y los beneficios asociados al mismo:

CONCEPTO	COSTE
Costes de Personal	31.500 Euros
Costes de Material	700 Euros
Riesgos y costes indirectos (10%)	3.220 Euros
Beneficio (0%)	0 Euros
TOTAL	35.420Euros

Tabla 35: Presupuesto Final

Por tanto, el presupuesto total para el desarrollo del proyecto es de **35.420 Euros**.

Referencias

[**Darwin, 1859**] Darwin, C. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, London, (1), 1–556.

[**Mitchell, 1996**] Mitchell, M. (1996) *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

[**Rechenberg, 1973**] Rechenberg, I. (1973). *Evolutionssstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Ph.D. thesis, TU Berlin.

[**Holland, 1975**] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

[**Mendel, 1865**] Gregor Mendel. (1865). *Experiments in Plant Hybridization*. Read at the February 8th, and March 8th, 1865, meetings of the Brünn Natural History Society.

[**JGAP**, <http://jgap.sourceforge.net/>]

[**ECLIPSE**, <http://www.eclipse.org/>]

[**UBUNTU**, <http://www.ubuntu.com/>]

